Introduction to LIS and LDT

Sujay Kumar and Kristi Arsenault

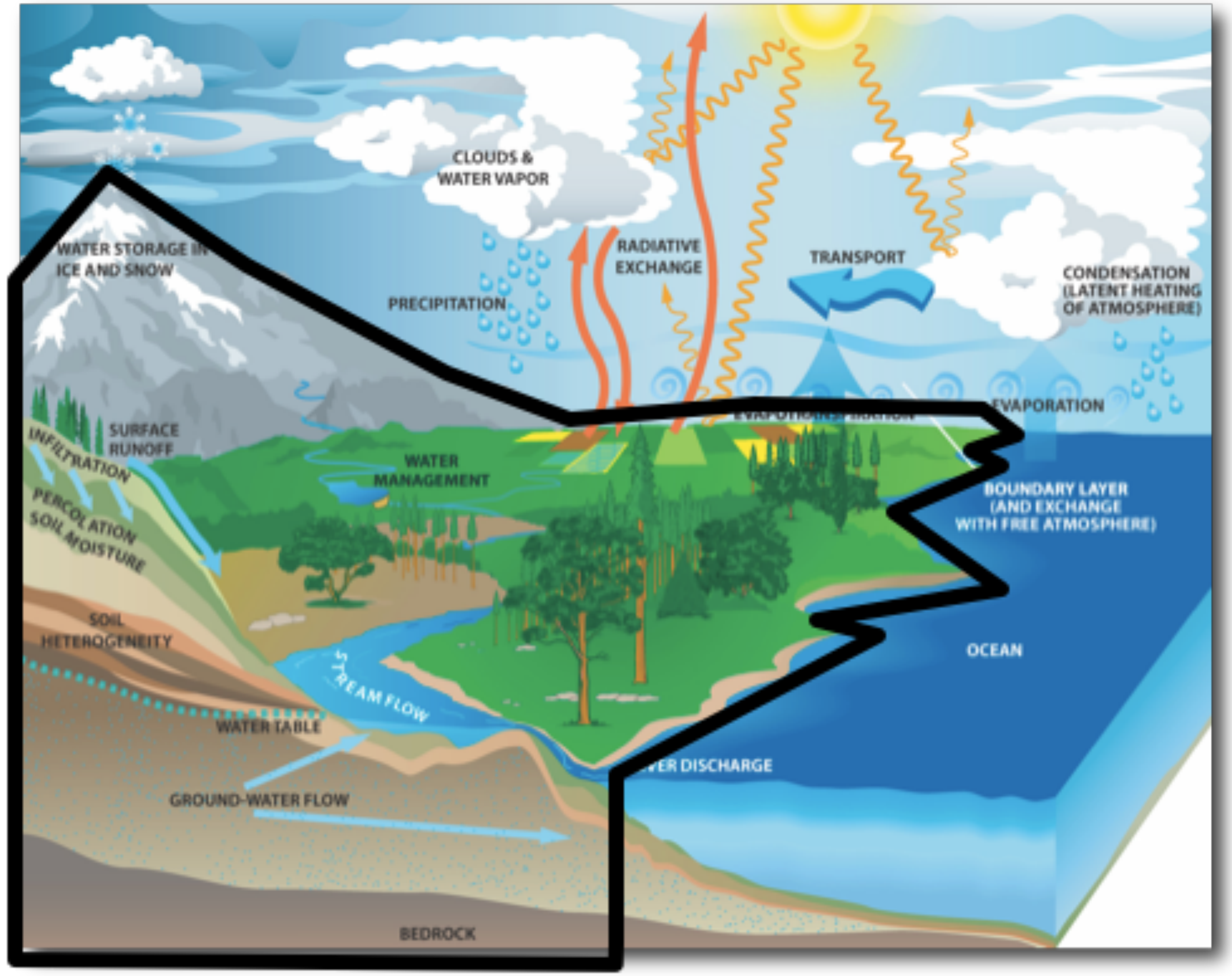
Science Applications International Corporation (SAIC) and Hydrological Sciences Laboratory, NASA/GSFC

LIS Tutorial, July 9-11, 2014

Outline

- What is LIS? – motivation, heritage
- Software architecture, design paradigms
- Source code, repository, software requirements
- LIS version 7 – upcoming features
- Two new toolkits that accompany LIS – LDT and LVT.

- A system to study land surface processes and land-atmosphere interactions
- “Use best available observations” to force and constrain the models
- Applications: Weather and climate model initialization, water resources management, natural hazards management

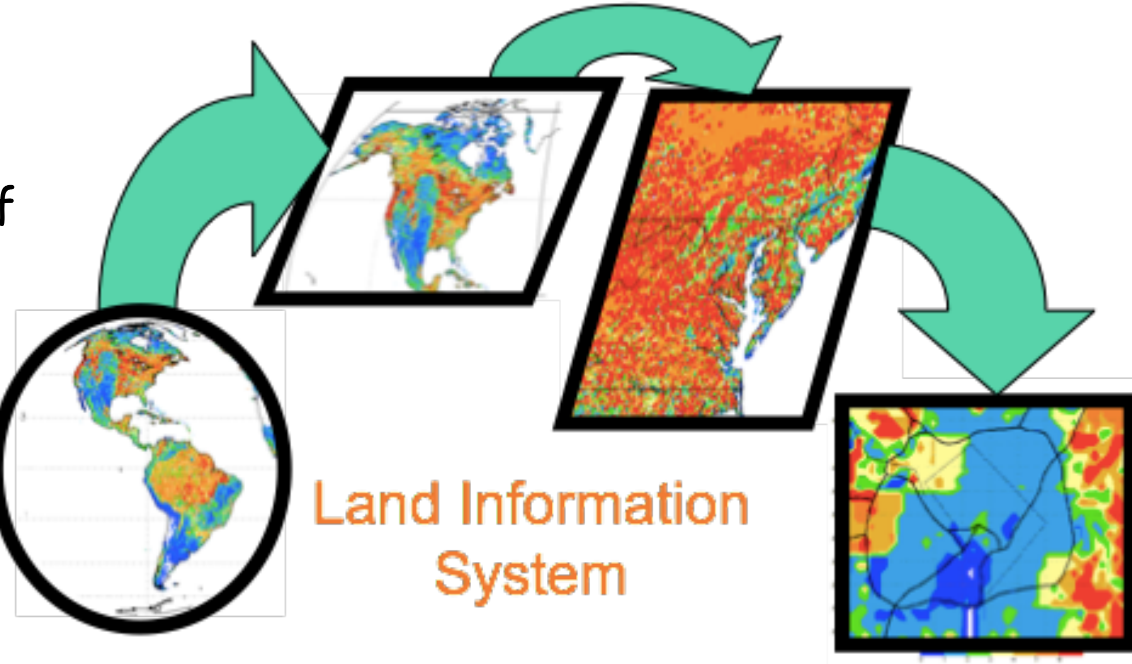


What is LIS?

- Need a system viable at different spatial and temporal scales

- Be able to demonstrate the impact of observations at the scale of observations themselves

- Explicit characterization of the land surface at the same spatial scales as that of cloud and precipitation processes helps in improving the characterization of land-atmosphere interactions

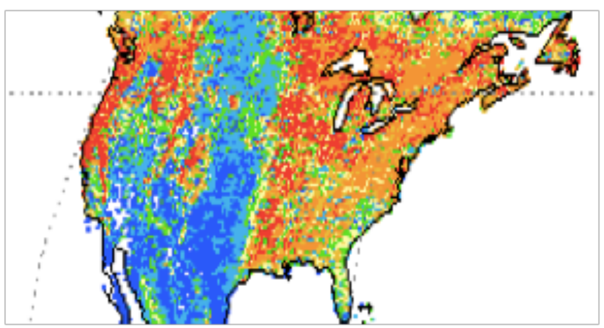


- Need scalable, high performance computing support to deal with computational challenges

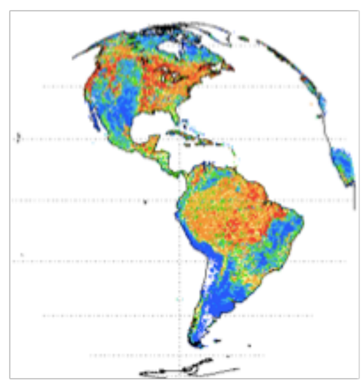
- Need advanced land surface models and modeling tools (data assimilation, optimization, uncertainty modeling)

LIS - heritage

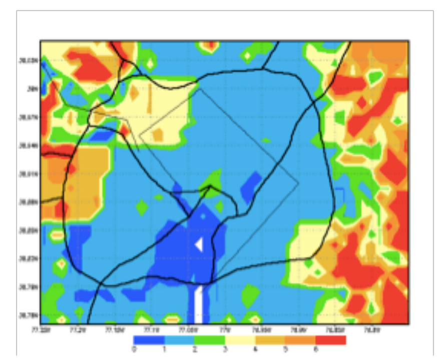
- LIS is a land surface modeling and data assimilation system (LDAS)
- Capable of modeling at different spatial scales, globally and regionally



North American LDAS
1/8th degree spatial resolution



Global LDAS
1/4th degree spatial resolution

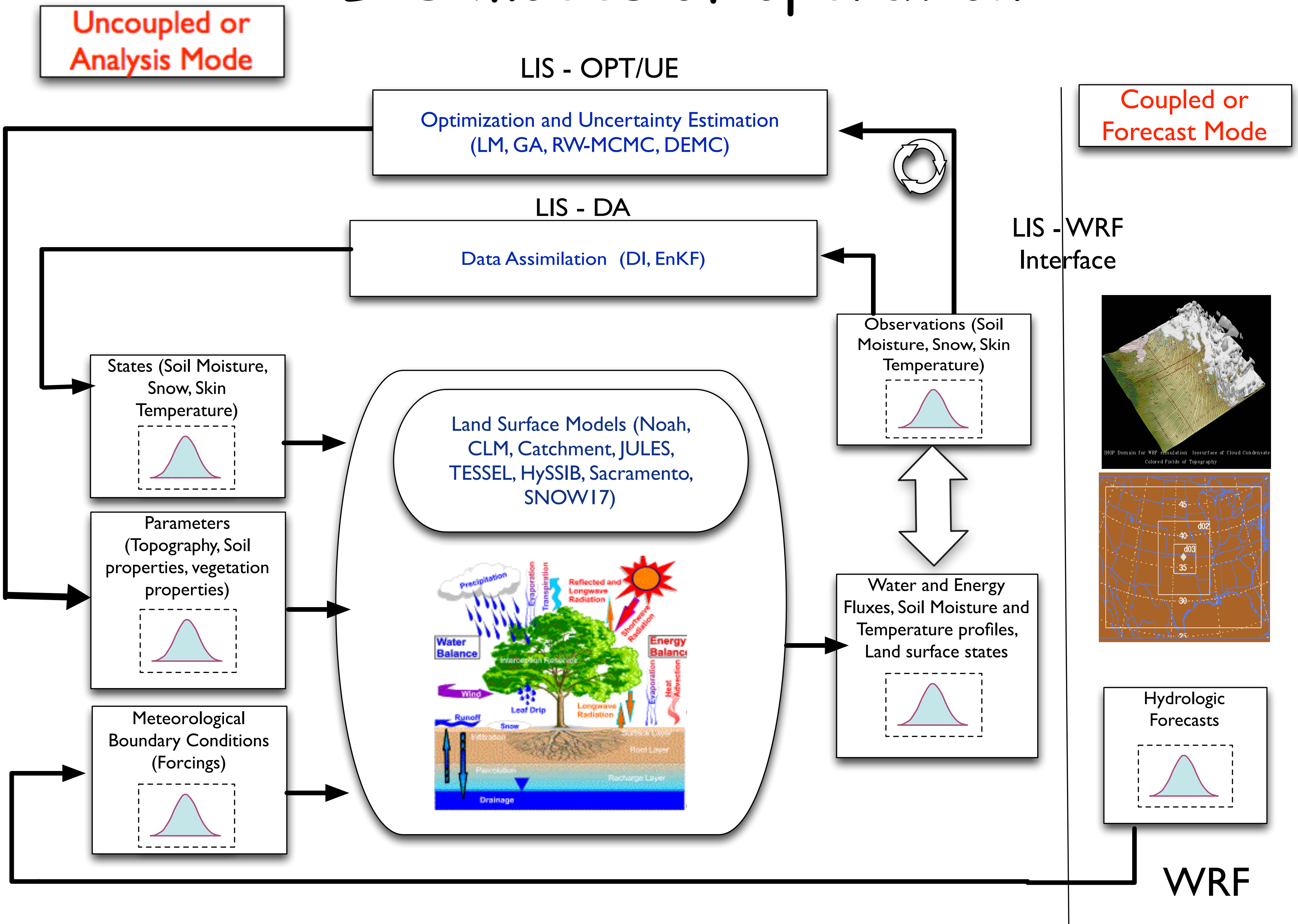


LIS
global, regional, point
up to 1km and finer

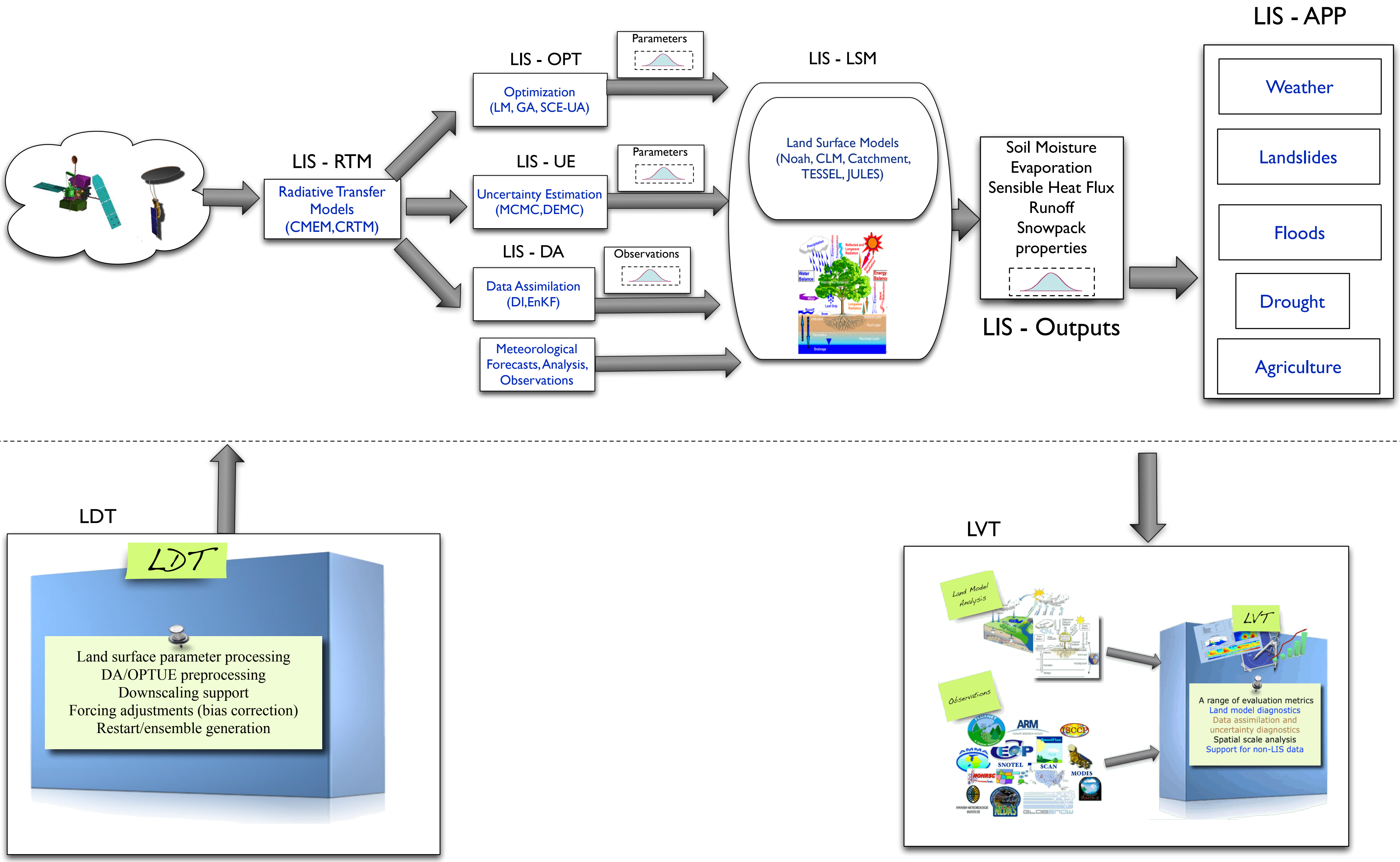
Kumar et al. (2006): Land Information System: An interoperable Framework for High Resolution Land Surface Modeling, Environmental Modeling and Software, Vol 21, pp 1402-1415.

Peter-Lidard et al. (2007): High-performance earth system modeling with NASA/GSFC's Land Information System, Innovations in Systems and Software Engineering, 3(3),157—165.

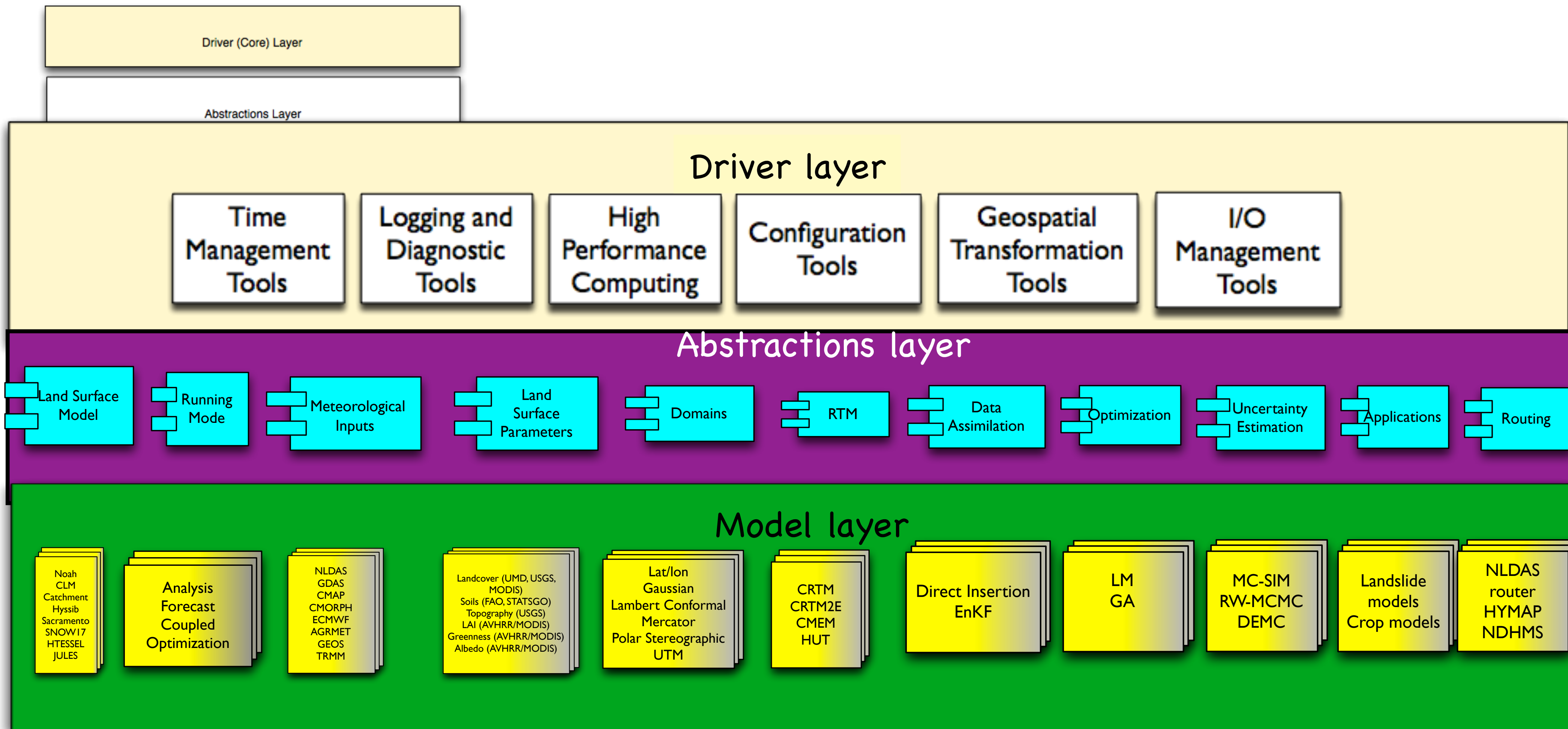
LIS modes of operation



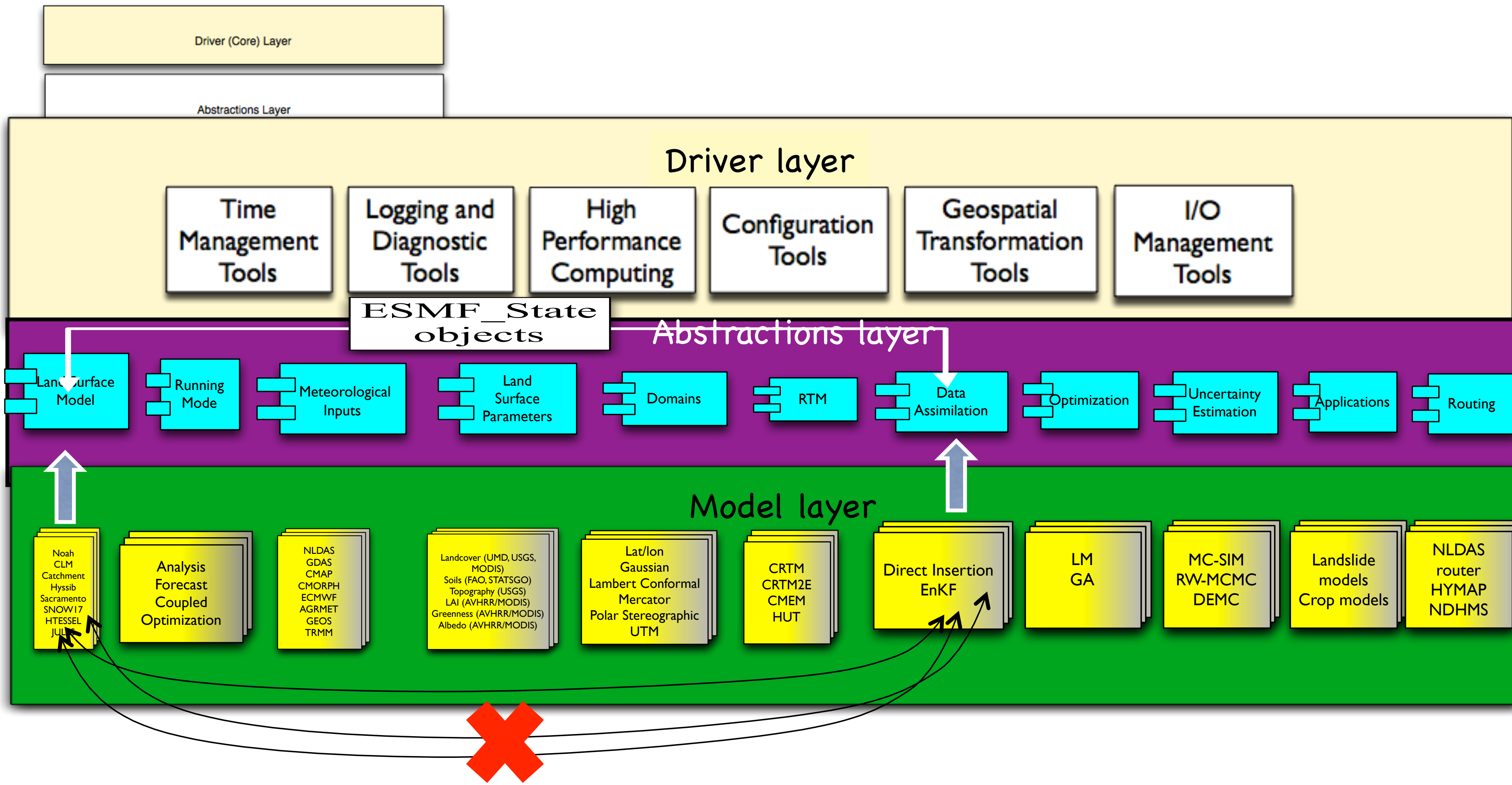
LIS subsystems and toolkits



LIS software architecture

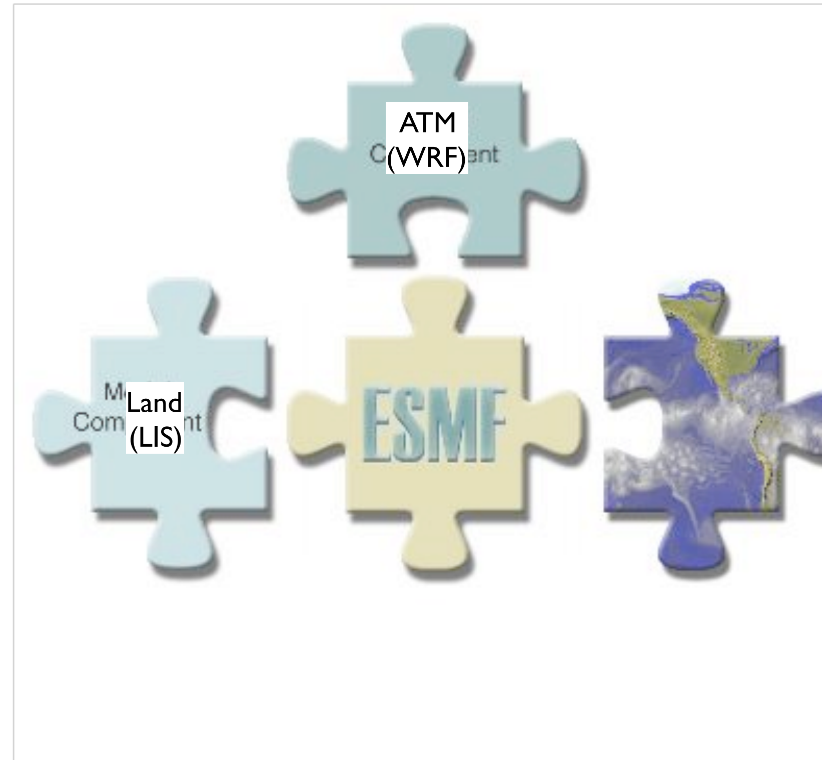


LIS software architecture



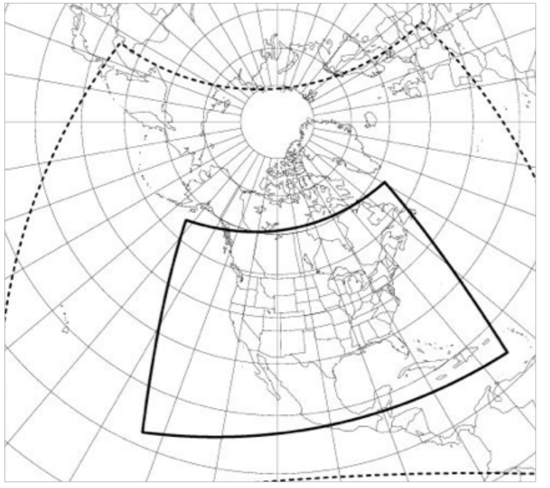
What is ESMF?

- Software for building and coupling weather, climate and related models
- Provides representations of Earth system grids, tools for mapping between them in multiprocessor environment
- Includes toolkits for building applications: time manager, error handling, resource management, parallel communications
- An application once it is wrapped with ESMF is known as a “Gridded component”
- Gridded components are coupled using “coupler components”



Key ESMF objects

- ESMF_Grid - representation of a grid
- ESMF_State - objects that hold gridded data
- ESMF_State consists of ESMF_Field, ESMF_Bundle, ESMF_Array
- Data is exchanged between ESMF Gridded components using ESMF_States



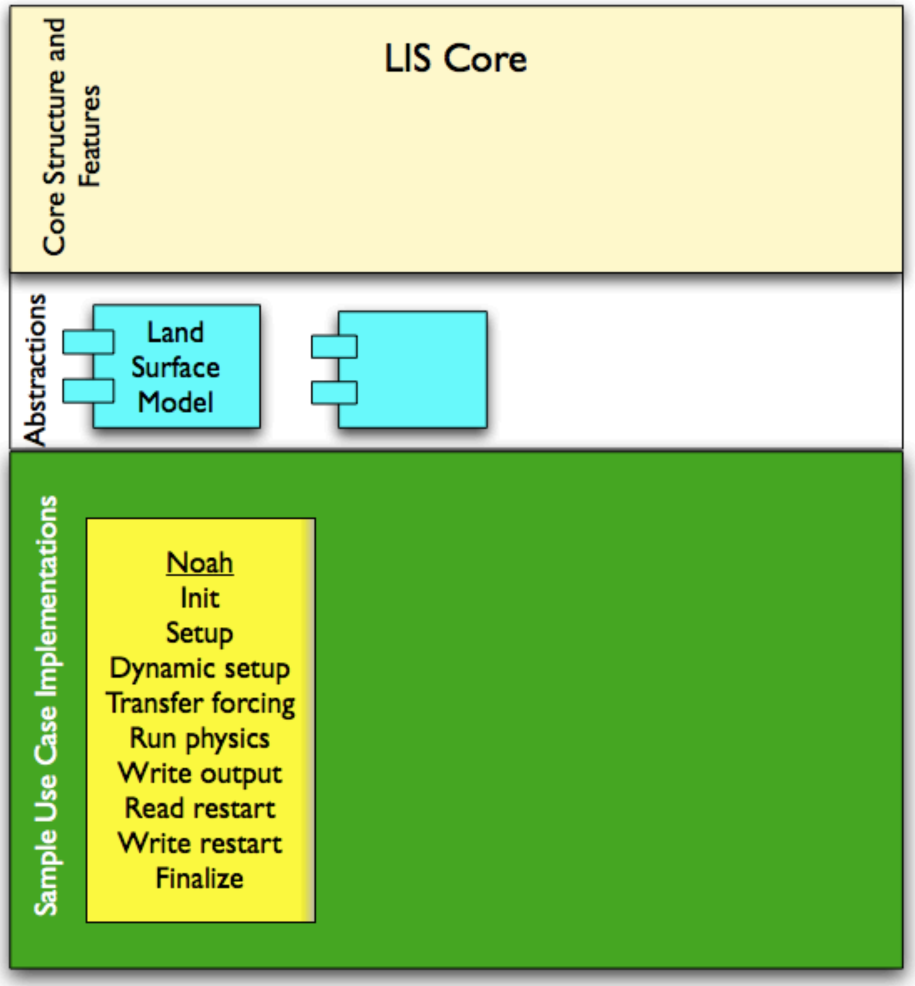
Object Oriented Programming

- Think Objects
- Modularity : Source code for an object, written and maintained independent of the source code for other objects
- Reusability: if the object already exists, you can use that object in your application
- Extensibility: Can be customized for new applications
- Inversion of Control - “Don’t call us, we’ll call you”
- Generic code controls execution of problem-specific code

States and Behavior

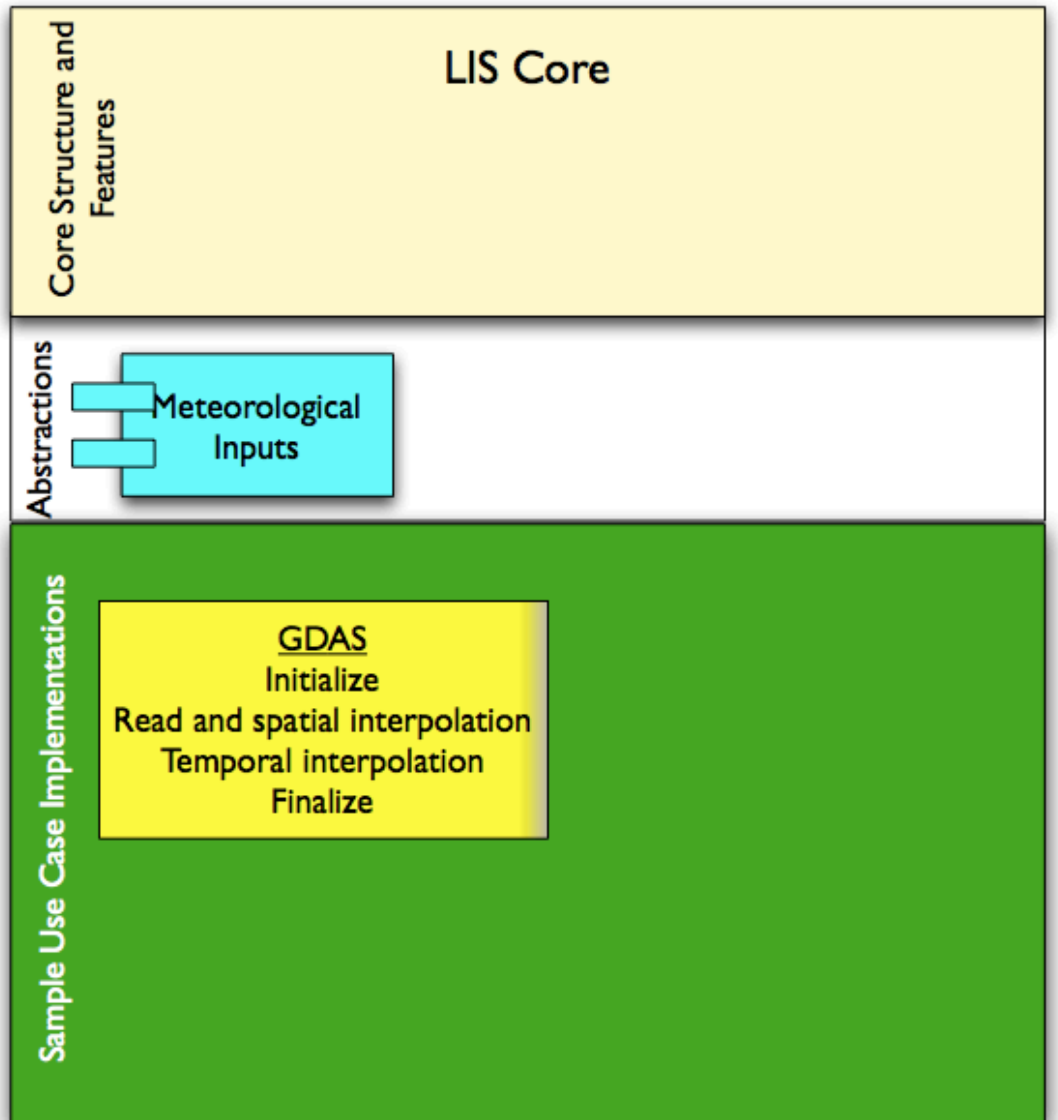
Gear
Speed
Pedal stance

Customization: How do we add a new LSM?



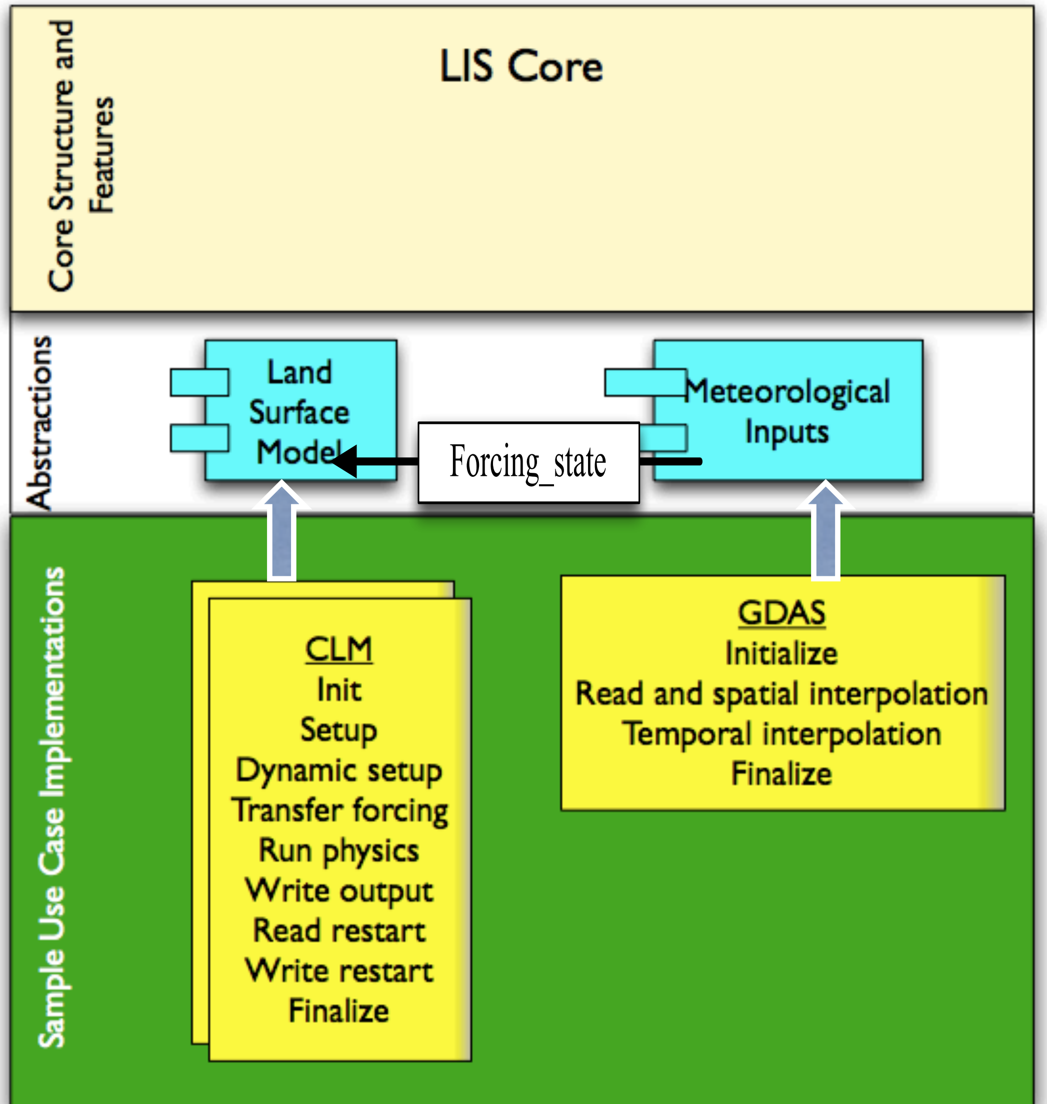
- Need to implement a set of interfaces related to the operation of a land surface model
- In LIS, these abstract implementations are known as "plugins"
- under src/plugins

How do we add a new forcing scheme?



- Extend the abstract interfaces related to a forcing scheme

Combining these components



LIS provides the "wirings" between the abstract implementations

Incorporating these components through plugins automatically ensure their integrated and interoperable use

LIS source code

http:://lis.gsfc.nasa.gov http:://modelingguru.nasa.gov

Software Requirements

- ☑ Fortran 90/95 compiler (g95 will not work for LIS5.0)
  - ☐ preferred : intel, pgi, lahey, absoft
- ☑ C compiler
- ☐ MPI - if parallel processing capability is desired
- ☑ Earth System Modeling Framework (ESMF)
  - ☐ 5.x – for LIS 7.0
- ☐ LIS supports Grib1, NETCDF, HDF formats
  - ☐ NETCDF- mandatory, Grib, HDF optional

LIS Documentation

- User's guide
  - Step-by-step instructions on how to build the LIS code
- Developer's guide
  - Instructions on how to bring in new functionalities (LSMs, forcing schemes, Data Assimilation, parameter data, etc.)
- Reference manual

LIS User's Guide

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

LIS Developer's Guide

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

Increasing Interoperability and Performance of
Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

May 7, 2004

Version 3.0

History:

Revision	Summary of Changes	Date
3.0	Milestone "G" submission	May 7, 2004
2.3	LIS 2.3 code release	December 19, 2003
	Initial revison	

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

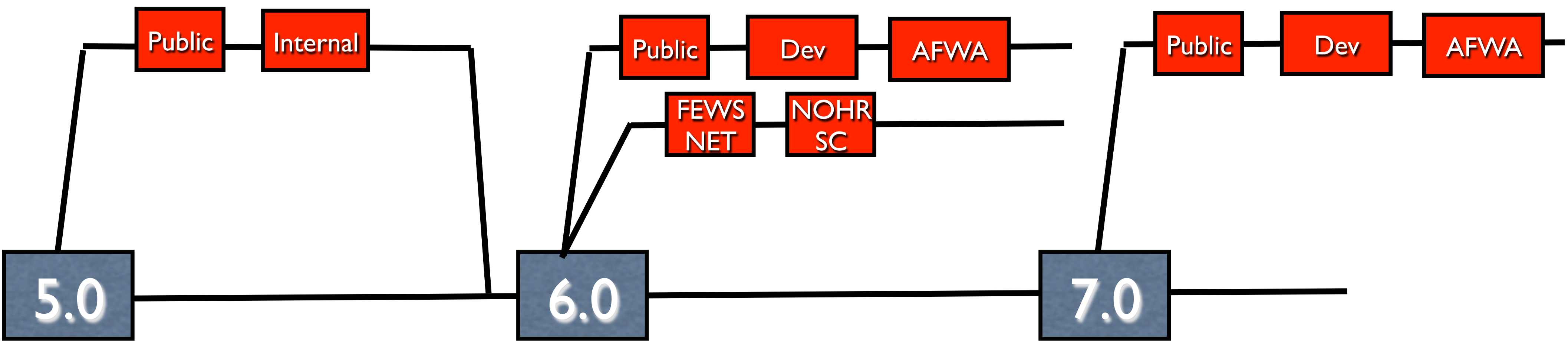
Getting LIS source

- Use the subversion repository (https://progress.nccs.nasa.gov)
- Apply for an account

  (christa.d.peters-lidard@nasa.gov)
- Request a "Project Release" of the LIS code (http://lis.gsfc.nasa.gov/register.shtml)
- Download the tarball (Internal users who use the repository can check out the appropriate branch)

LIS source code repository

- Helpful links
  - http://subversion.tigris.org/
  - http://svnbook.red-bean.com/
- Check out the LIS code:
  - svn co https://progress.nccs.nasa.gov/svn/lis/ src



The Land surface Data Toolkit (LDT): Overview and Examples

The “preprocessor” to LIS-7

July 9, 2014

The Land Data Toolkit (LDT)

- A new preprocessing toolkit for LIS's model parameters
  - Processes and groups parameters needed for each land surface model (LSM)
  - Multiple processing options
- Observation-based data assimilation options
  - CDF-matching, etc.
- Generate custom-made restart files for LSMs
  - e.g., ensemble-based restart files

LDT Parameter Preprocessing

- Offers subsetting, reprojecting, aggregating/downscaling and new tiling options for parameter files onto the targeted LIS-7 grid and domain;
- Group parameter files together in to a common Netcdf 4 file, which includes header and grid information for the data layers;
- This grouped netcdf file can then be easily loaded and read by LIS-7 at run-time;
- Building into LDT also LSM run-time checks (e.g., correct parameters selected) and parameter statistics (e.g., land/water mask gridcell effects)

The Land Data Toolkit (LDT): for Parameter Processing

Parameter Inputs:

- Landmask
- Landcover
- Elevation
- Soils data
- LSM Parms:
  - Noah LSM
  - VIC
  - Catchment LSM
  - Others …

Domain Information:

- LIS Target Domain and Projection
  - e.g., NLDAS domain
- Individual Parameter Grid Info

LDT Output:

- LIS Target Domain and Projection
- One netcdf file that gets read by LIS
- Netcdf files can be generated for each model, set of parameters, etc.

LDT DA Observations Processing

For DA preprocessing, 3 options are supported:

- Generate cumulative density functions (CDFs) ➔ for CDF scaling of observations within LIS;
- Generate mean and standard deviations for normal-deviate based scaling within LIS;
- Anomaly correction to observations (for GRACE-DA primarily).

LDT Restart File Generation

(featuring ensemble restart generation)

- For ensemble restarts, two current options:
  - *Upscaling*: going from one to many ensemble members, and
  - *Downscaling*: going from many members to one.
- Future restart generation options may include:
  - Climatological restart generation,
  - Spatial aggregation/downscaling of existing restart files

Other LDT features (as originally in LIS)

- Writes run-time diagnostic information to a log file, like in LIS;
- Handle different LIS projections (all *interp* routines as available in LIS):
  - Lat-lon, Lambert, Polar stereographic, etc.
- Work with multiple member ensembles;
- Working toward reading in all original LSM parameter files (of LSMs available in LIS-7);
- Run multiple domain nests
- *Future additions: Parallel processing …*

Current LDT Capabilities

1. Still able to read in original "LIS-produced" parameter and observation files with binary format of:
   - Direct-access, 4-byte real, big-endian binary …
   - Lat-lon gridded files with resolutions starting at 0.01° (1-KM), which can be aggregated to other coarser resolutions and projections;
2. Moving towards reading in all "native" or "raw" parameter files, obtained from the original institutional source
3. Checks performed on parameter file processing;

   -- e.g., Ensure land cover and mask match correctly; (sand+silt+clay) fractions == 1

   -- e.g., Parameter-landmask land/water agreement.

Current Features

Main LDT Configuration File Options

LDT Main Run Modes

LDT Option	Option	Description
LDT Running Mode:	*LSM parameter processing*	Processes the parameters required for a given model run within LIS and outputs to a common netcdf file
	DA preprocessing	Prepares certain bias correction statistics and observation processing for data assimilation (DA) runs in LIS
	Ensemble restart processing	Generates ensemble restart files, e.g., netcdf-4 file with header info
LSM parameter attributes file:	*e.g., param_attribs.txt*	Parameter attributes file which contains many parameter label and dimension options
Processed LSM parameter filename:	*e.g., lis_input.d01.nc*	Generated LSM parameter file in netcdf-4 format, required for running LIS-7.

LIS Output Grid Options

LDT Option	Option	Description
Map projection of the LIS domain:	*latlon, lambert, polar, hrap, gaussian, mercator*	Different output projection options on which to write the parameters for a LIS experiment.
Lat/Lon Rectangular Grid Dimensions (Extents)		
Run domain lower left lat/lon:	*e.g., 40.125; -94.875*	Enter the lowest-left corner extent latitude and longitude values for your LIS target grid
Run domain upper right lat/lon:	*e.g., 49.875; -75.125*	Enter the upper-right corner extent latitude and longitude values for your LIS target grid
Run domain resolution (dx/dy)	*e.g., 0.25; 0.25*	Enter the x- and y-direction gridcell resolution for your LIS target grid

The *LDT.config* file

```
LDT running mode:              "LSM parameter processing"  # LDT type of run-mode (top-level option)
LSM parameter attributes file:    ./param_attribs.txt    # List of LSM Parameter types
Processed LSM parameter filename:  ./lis_input.d01.nc     # Final output file read by LIS-7

LIS number of nests:                   1                 # Total number of nests run by LIS
Number of surface model types:         1                 # Total number of desired surface model types
Surface model types:                 "LSM"               # Surface models:  LSM | Openwater
Land surface model:                  "Noah.3.3"          # Enter LSM(s) of choice
Lake model:                          "none"              # Enter Lake model(s) of choice
Water fraction cutoff value:          0.5                # Fraction at which gridcell is designated as 'water'

Number of met forcing sources:         0                 # Enter number of forcing types
Met forcing sources:                 "none"              # Enter 'none' if no forcing selected
Met spatial transform methods:      bilinear             # bilinear | budget-bilinear | neighbor | average
Topographic correction method (met forcing):  "none"     # none | lapse-rate

LDT diagnostic file:                ldtlog               # Log-based diagnostic output file
Undefined value:                   -9999.0               # Universal undefined value
Metrics output directory:           OUTPUT               # If metrics or stats are written out
Number of ensembles per tile:         1                  # The number of ensemble members per tile

# Processor layout (currently not available)
Number of processors along x:       1
Number of processors along y:       1

# LIS domain:  (See LDT User's Guide for other projection information)
Map projection of the LIS domain:     latlon
Run domain lower left lat:           25.0625
Run domain lower left lon:         -124.9375
Run domain upper right lat:          52.9375
Run domain upper right lon:         -67.0625
Run domain resolution (dx):           0.125
Run domain resolution (dy):           0.125
```

Land cover and mask options

LDT Option	Option	Description
Land cover classification:	*UMD, IGBPNCEP, USGS, and others*	Enter the land cover/land use classification scheme, which varies with different sources
Create or readin landmask:	*create*	Creates a landmask from the landcover file
	readin	Reads in a landmask file; future releases will include more options to impose mask on parameter files
landcover map projection:	*latlon, and others (depending on the parameter read in)*	Projection/grid type for the input land cover parameter file.
landcover spatial transform:	*e.g., tile, mode, nearest neighbor*	Mode for transforming the input parameter projection/grid to the LIS output (run-time) grid
landcover [grid-domain inputs]:	*For example,* *Landcover lower left lat: -59.995* *Landcover lower left lon: -179.995*	Similar to how LIS run domain extents were specified

The *LDT.config* file (con't.)

Original LIS-team produced data entries:

```
# Landcover/Mask Parameter Inputs
Landcover classification:     "IGBPNCEP"        # Enter land cover classification type
Landcover file:      ../noah_lisparms_testcase/UMD/1KM/landcover_IGBP_NCEP.1gd4r  # Landcover map path
Landcover spatial transform:    tile            # none | mode | neighbor | tile
Landcover fill option:          none            # none | neighbor (Not needed if creating landmask)
Landcover map projection:       latlon
# For "LIS" based files, must enter grid info:
Landcover lower left lat:      -59.995          # For "LIS" based files, must enter grid info
Landcover lower left lon:     -179.995
Landcover upper right lat:      89.995
Landcover upper right lon:     179.995
Landcover resolution (dx):       0.01
Landcover resolution (dy):       0.01
```

"Native" or raw data entries:

```
# Landcover/Mask Parameter Inputs
Landcover classification:       "IGBPNCEP"        # Enter land cover classification type
Landcover file:               ./input/igbp.bin    # Landcover map path
Landcover spatial transform:      tile            # none | mode | neighbor | tile
## Note: The fill entries here will 'fill' a missing landcover value with a neighboring one or assigned value.
Landcover fill option:          neighbor          # none | neighbor (Not needed if creating landmask)
Landcover fill radius:           3.               # Number of pixels to search for neighbor
Landcover fill value:            5.               # Static value to fill where missing
Landcover map projection:       latlon            # Landcover map projection
```

The *LDT.config* file (con't.)

Read in Landmask entries:

```
# Create landmask field from readin landcover map or read in separate landmask file
Create or readin landmask:      "readin"               # create | readin
Landmask file:             ./input/landmask_UMD.1gd4r  # Land mask file (if needed to be read-in)
Landmask spatial transform:       mode                 # none | mode | neighbor
## Note: The 'mode' entry here will determine the dominant mask value for each gridcell. ##
Landmask map projection:         latlon                # Landmask map projection
Landmask lower left lat:        -59.995                # Extents of the 1KM UMD landmask
Landmask lower left lon:       -179.995
Landmask upper right lat:        89.995
Landmask upper right lon:       179.995
Landmask resolution (dx):         0.01
Landmask resolution (dy):         0.01
```

"Creating" Landmask from Landcover entries:

```
# Create landmask field from readin landcover map or read in separate landmask file
Create or readin landmask:    "create"          # create | readin
Landmask file:                  none            # Land mask file (if needed to be read-in)
Landmask spatial transform:     none            # none | mode | neighbor
Landmask map projection:       latlon
```

Current LDT Processing of Landcover and Masks

Two Major User-Given Options

1. Parameters on <u>SAME</u> grid/ resolution as LIS target output grid

1. Can read in *landcover and mask* and subset to desired LIS target output grid (current approach in LIS-6)
2. Apply mask to subsequent routines and parameters

2. Parameters on <u>DIFFERENT</u> grid/ resolution as LIS target output grid

1. Read in *landcover* (e.g., @ 1 km res.) and aggregate or interpolate to LIS target grid
2. Create new land/water mask from new mapped-to-LIS-grid land cover field
3. Apply mask to subsequent routines and parameters

Spatial Transform Options

LDT Option	Option	Description
[Parameter] spatial transform*:	**none**	No scaling or transformation needed.
	average	Average finer resolution grid to coarser scale one (*upscale).*
	mode	Select dominant (mode) of finer scale gridcells to coarser one (*upscale*).
	tile	Apply accounting system for estimating coarser sub-grid "tiles" for output grid.
	neighbor	Apply nearest neighbor method to input grid (*upscale/downscale*).
	bilinear	Apply bilinear interpolation method to input grid (*upscale/downscale*).
	budget-bilinear	Apply conservative "budget" bilinear interpolation method to input grid (*upscale/downscale*).
	**** NOTE: Not all options available yet for all parameters.***	

Fields that can be tiled (or binned) …

LDT Option	Option	Description
Land cover	*Number of vegetation tiles*	Derives fractions for vegetation type-tiles (summing to 100%)
Lake surface	*Number of lakes (as relates to FLake at this time)*	Lake fractions can be estimated from lake depth or lake type
Soil fractions	*Fraction of sand, silt and clay*	Soil fraction bins (or tiles) along with average fractions per bin
Soil texture	*Soil texture tiles*	Derives fractions for soil texture types
Elevation	*Elevation tiles (aka, "bands")*	Average elevations associated with ranked tiles or "bands"
Slope, Aspect	*Tile fractions and associated average slope or average aspect*	Average slope or aspect associated with ranked tiles

The ***param_attribs.txt*** file example

LIS-team processed (aka, “LIS-data”) parameters

```
Landmask:          1  AVHRR        - 1  1 "AVHRR-(UMD) land mask"
Landcover:         1  MODIS_LIS    - 1 20 "MODIS-IGBP (LIS) land cover"
Soil texture:      1  STATSGOFAO_LIS - 1 16 "STATSGO+FAO-(LIS) soil texture"
Slope type:        1  NCEP_LIS     - 1  1 "NCEP-(LIS) slope type"
Albedo:            1  NCEP_LIS     - 12 1 "NCEP-(LIS) albedo climatology"
Max snow albedo:   1  NCEP_LIS     - 1  1 "NCEP-(LIS) max snow free albedo"
Greenness:         1  NCEP_LIS     - 12 1 "NCEP-(LIS) greenness climatology"
Shdmin:            1  NCEP_LIS     - 1  1 "NCEP-(LIS) greenness min"
Shdmax:            1  NCEP_LIS     - 1  1 "NCEP-(LIS) greenness max"
Tbot:              1  NCEP_LIS     K 1  1 "NCEP-(LIS) Bottom temperature"
```

The parameter attributes table (param_attribs.txt) allows the user to select different options for a given parameter type, including distinguishing between “LIS-data” and “Native” parameter data types.

LDT DA Obs Inputs

- **Inputs:**
  - Enter DA observation type (several options)
  - Specify the number of bins to estimate the CDF
  - Different temporal and spatial map (mask) options for CDF stats
  - Observation count in estimation

```
LDT running mode:                    "DA preprocessing"
DA observation source:               "RT SMOPS soil moisture"
Apply anomaly correction to obs:     0
Compute CDF:                         1
Number of bins to use in the CDF:    100
Name of CDF file:                    'rtsmops_cdf'
Temporal averaging interval:         "1da"
Apply external mask:                 0
External mask directory:             none
Observation count threshold:         400 #20 % of the record

RT SMOPS soil moisture observation directory: ../RT_SMOPS/
RT SMOPS soil moisture use ASCAT data:        1
```

LDT Examples

Highlighting new and unique features of LDT and LIS-7 inputs

Example NetCDF Output
(e.g., *lis_input.d01.nc*)

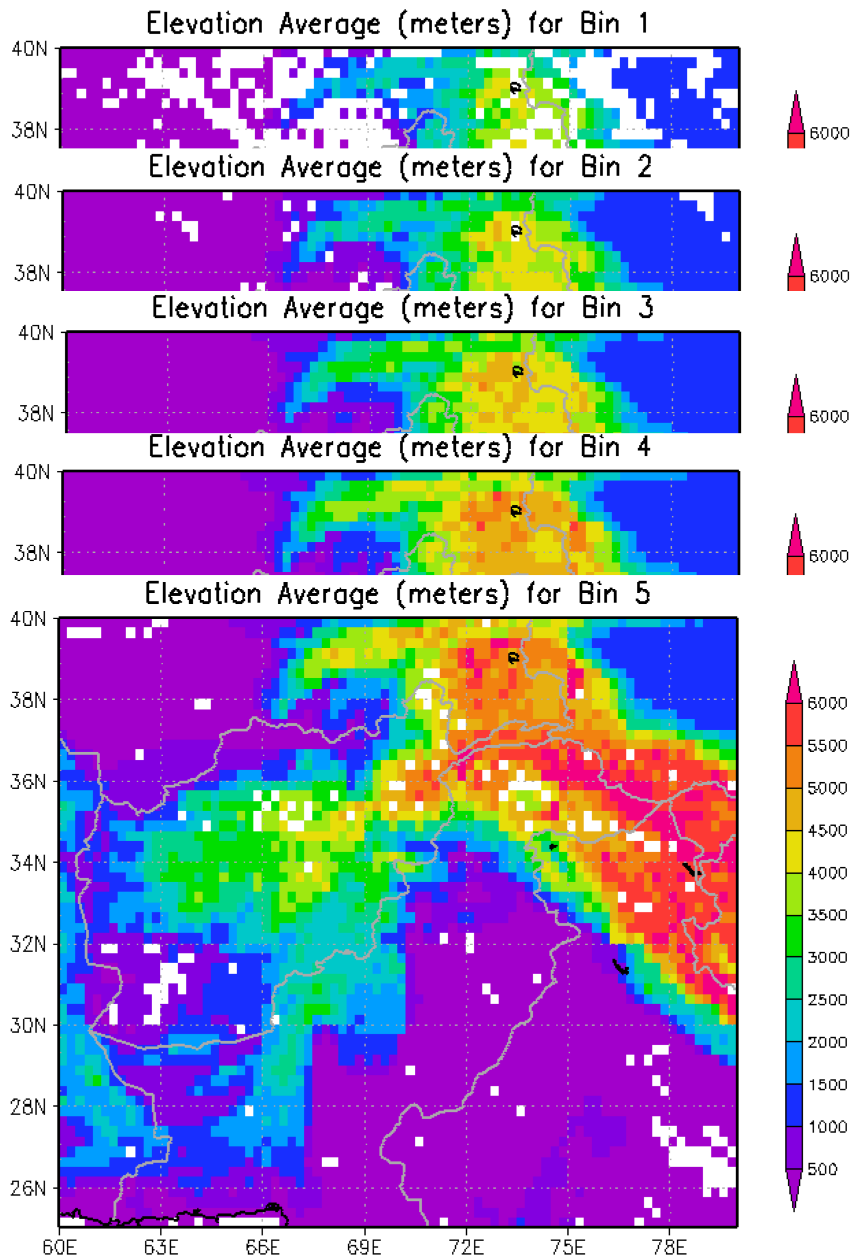
karsenau@discover15:...pt_working/lpt_run

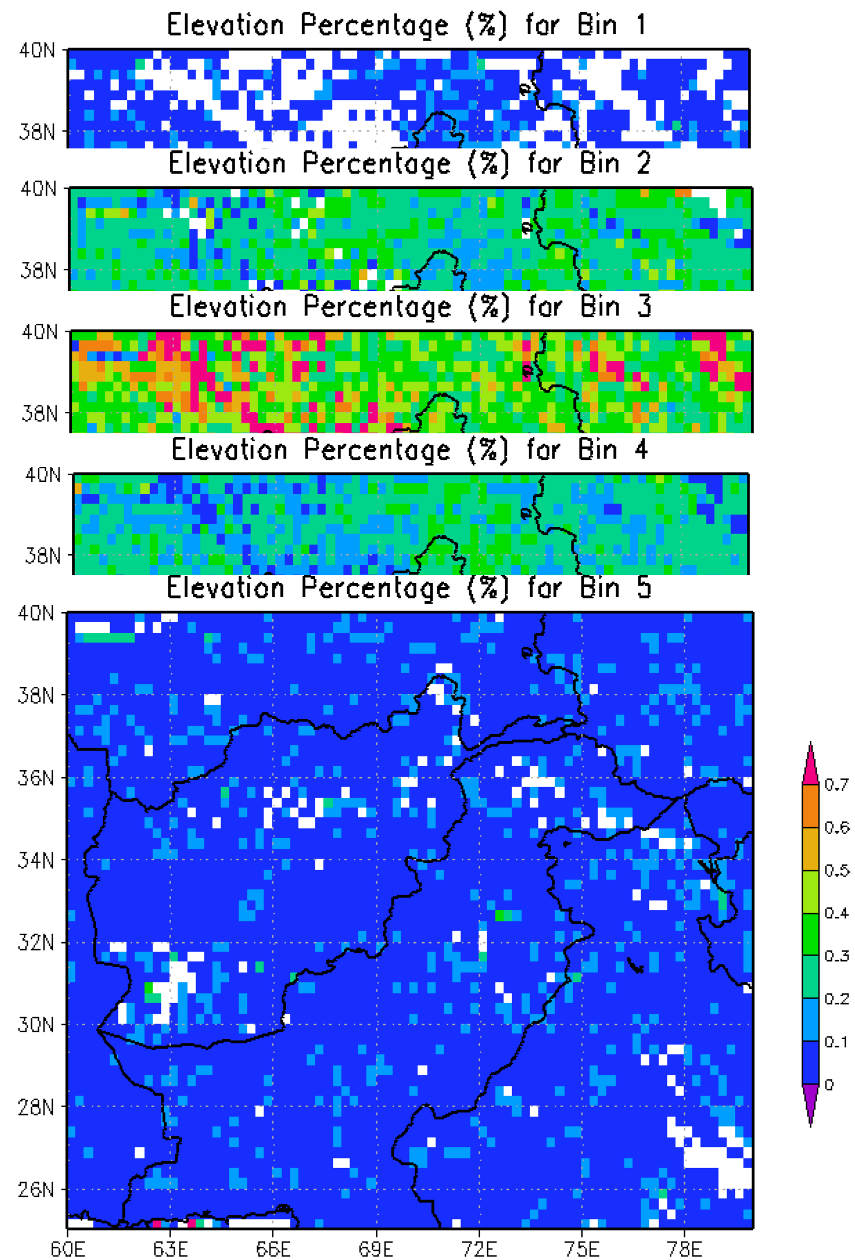
```
netcdf lis_input.d01 {
dimensions:
        east-west = 226 ;
        north-south = 110 ;
        vegtypes = 13 ;
        month = 12 ;
        time = 1 ;
variables:
        float time(time) ;
        float LANDMASK(north-south, east-west) ;
                LANDMASK:standard_name = "UMD land mask" ;
                LANDMASK:units = "-" ;
                LANDMASK:scale_factor = 1.f ;
                LANDMASK:add_offset = 0.f ;
                LANDMASK:missing_value = -9999.f ;
                LANDMASK:vmin = 0.f ;
                LANDMASK:vmax = 0.f ;
        float LANDCOVER(vegtypes, north-south, east-west) ;
                LANDCOVER:standard_name = "UMD land cover" ;
                LANDCOVER:units = "-" ;
                LANDCOVER:scale_factor = 1.f ;
                LANDCOVER:add_offset = 0.f ;
                LANDCOVER:missing_value = -9999.f ;
                LANDCOVER:vmin = 0.f ;
```

```
        float TEXTURE(north-south, east-west) ;
                TEXTURE:standard_name = "STATSGO soil texture" ;
                TEXTURE:units = "-" ;
                TEXTURE:scale_factor = 1.f ;
                TEXTURE:add_offset = 0.f ;
                TEXTURE:missing_value = -9999.f ;
                TEXTURE:vmin = 0.f ;
                TEXTURE:vmax = 0.f ;
        float SAND(north-south, east-west) ;
                SAND:standard_name = "FAO sand fraction" ;
                SAND:units = "-" ;
                SAND:scale_factor = 1.f ;
                SAND:add_offset = 0.f ;
                SAND:missing_value = -9999.f ;
                SAND:vmin = 0.f ;
                SAND:vmax = 0.f ;
        float CLAY(north-south, east-west) ;
                CLAY:standard_name = "FAO clay fraction" ;
                CLAY:units = "-" ;
                CLAY:scale_factor = 1.f ;
                CLAY:add_offset = 0.f ;
                CLAY:missing_value = -9999.f ;
                CLAY:vmin = 0.f ;
                CLAY:vmax = 0.f ;
        float SILT(north-south, east-west) ;
                SILT:standard_name = "FAO clay fraction" ;
                SILT:units = "-" ;
                SILT:scale_factor = 1.f ;
                SILT:add_offset = 0.f ;
                SILT:missing_value = -9999.f ;
                SILT:vmin = 0.f ;
                SILT:vmax = 0.f ;
        float ELEVATION(north-south, east-west) ;
                ELEVATION:standard_name = "GTOPO30 elevation" ;
                ELEVATION:units = "m" ;
                ELEVATION:scale_factor = 1.f ;
                ELEVATION:add_offset = 0.f ;
                ELEVATION:missing_value = -9999.f ;
                ELEVATION:vmin = 0.f ;
                ELEVATION:vmax = 0.f ;
        float GREENNESS(month, north-south, east-west) ;
                GREENNESS:standard_name = "NCEP greenness climatology" ;
                GREENNESS:units = "-" ;
                GREENNESS:scale_factor = 1.f ;
                GREENNESS:add_offset = 0.f ;
                GREENNESS:missing_value = -9999.f ;
                GREENNESS:vmin = 0.f ;
```

Parameter Tiling Examples

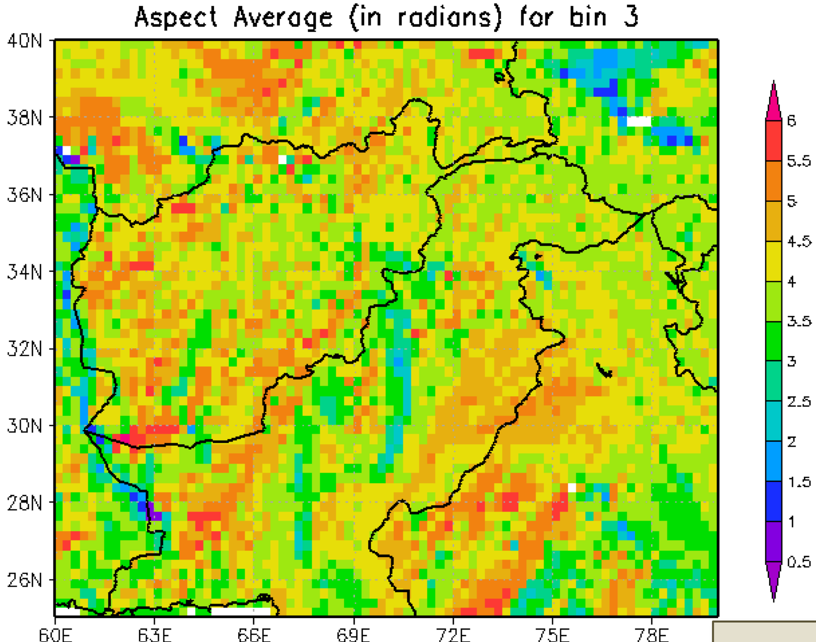
Elevation banding example (5 bins) for Afghanistan-Pakistan region (0.25 deg lat-lon grid)

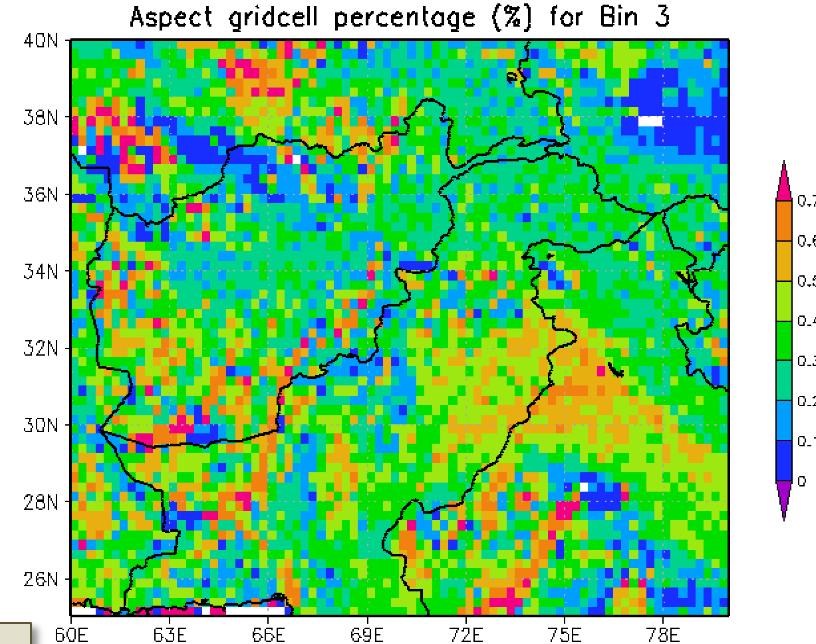




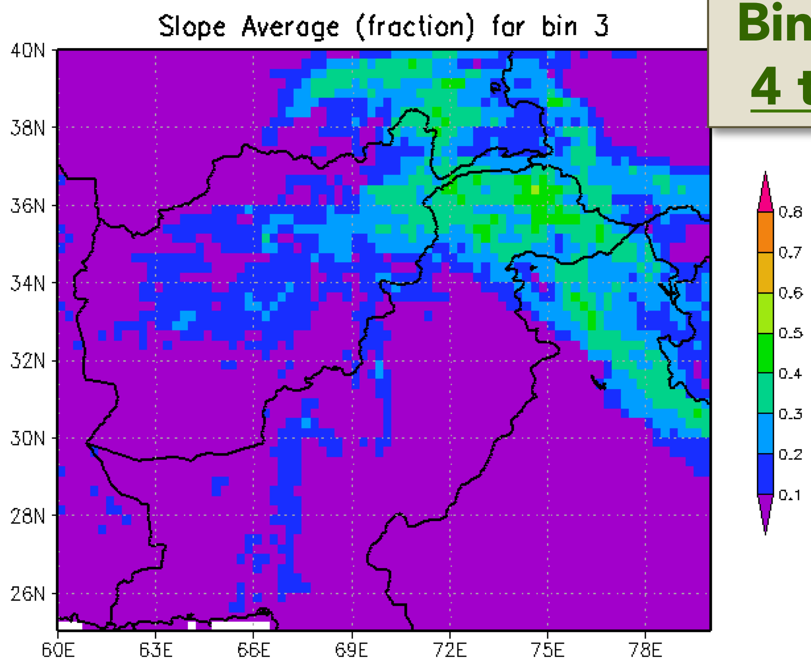
Slope and Aspect Tiling Examples

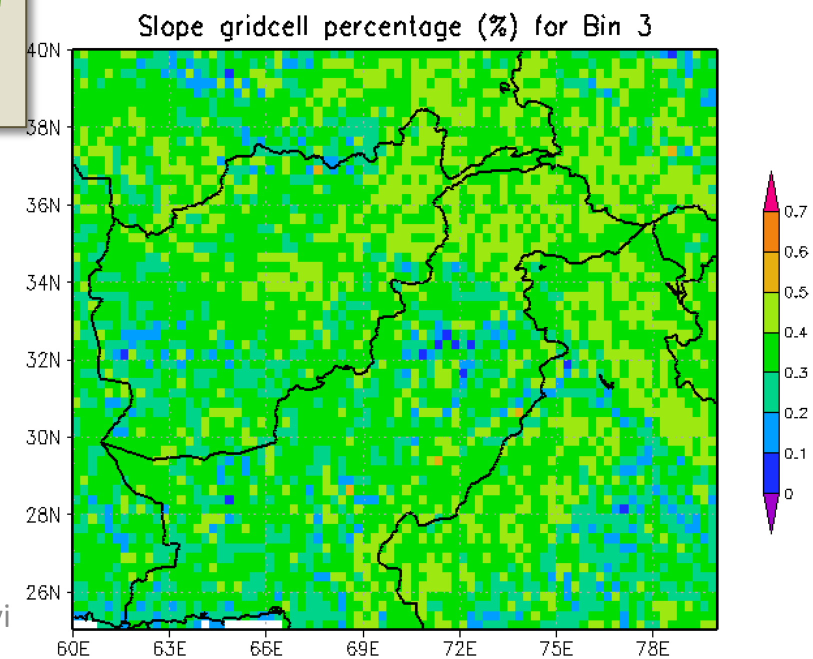
Slope, aspect banding examples, Afghanistan-Pakistan region (0.25 deg lat-lon grid)





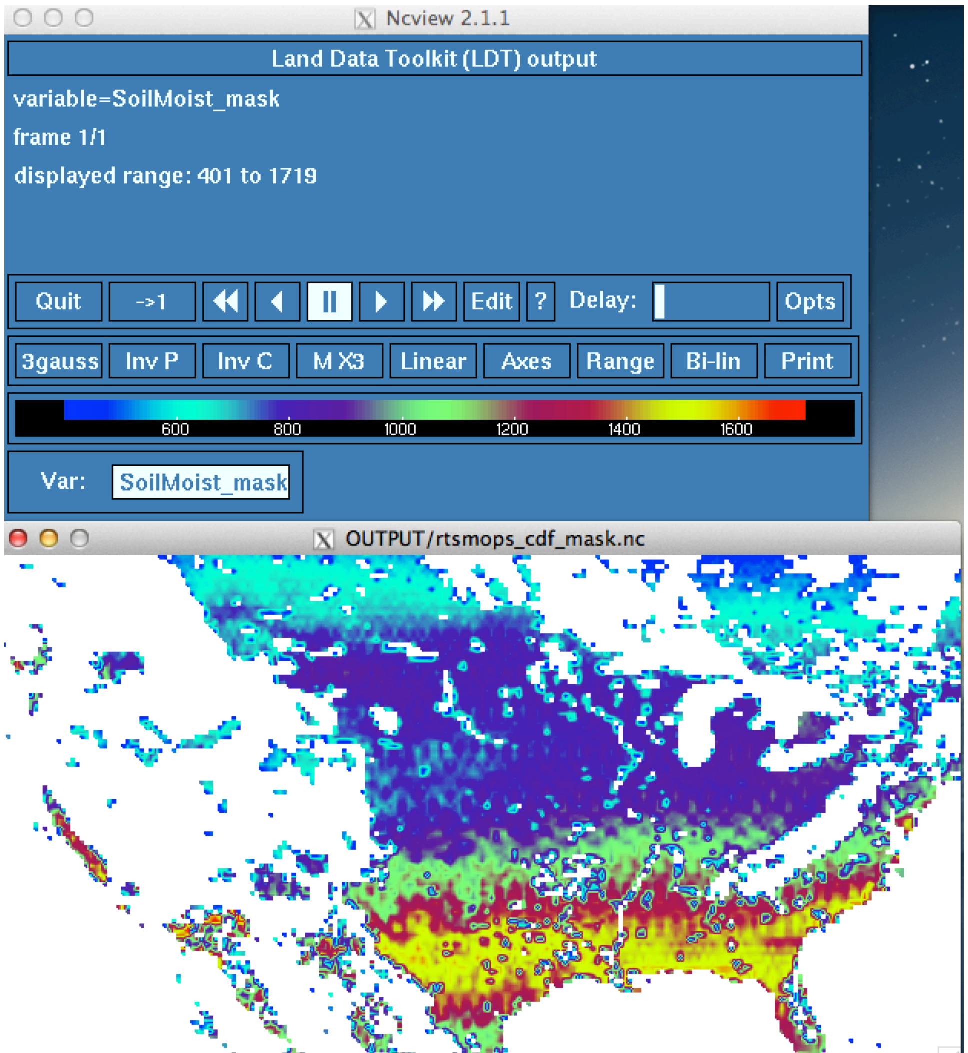
Bin 3 of <u>4 total</u>

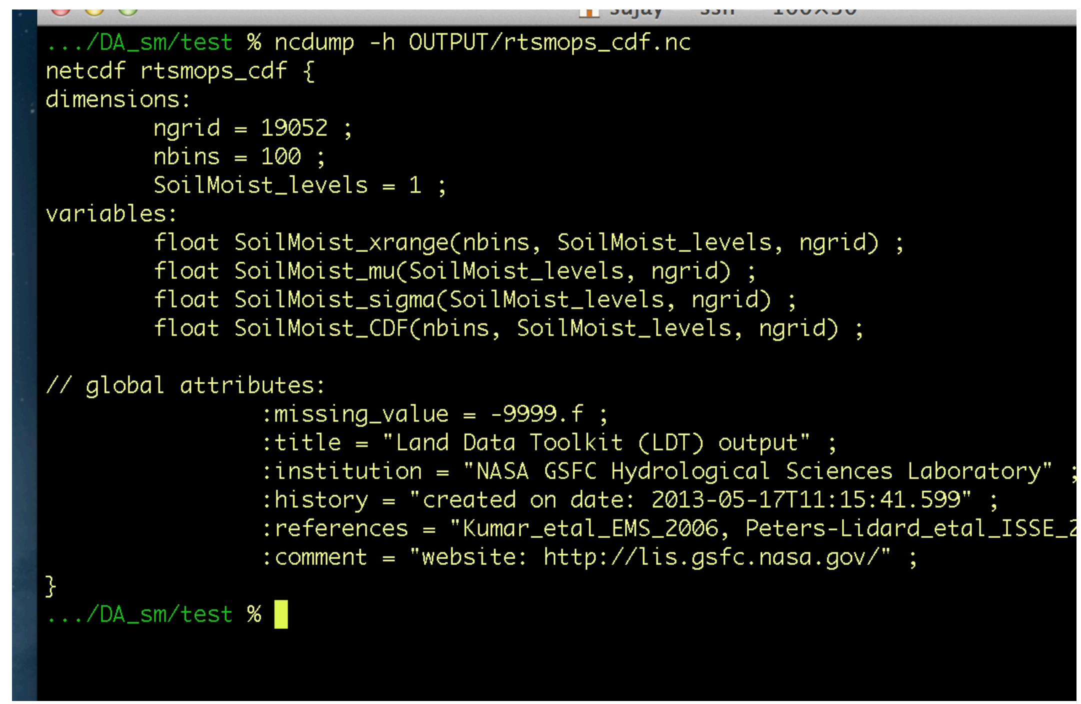




DA Observations Examples

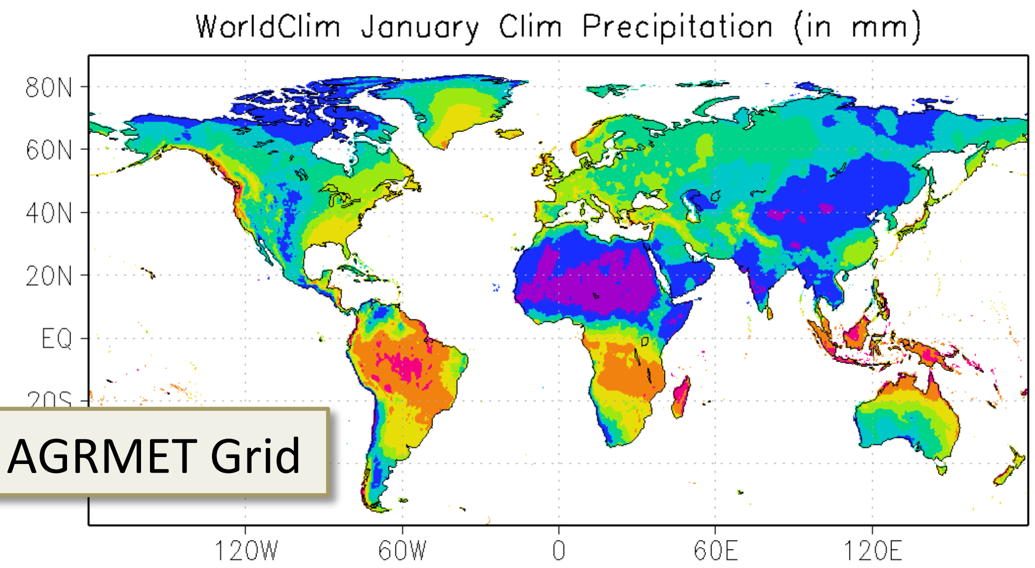
- LDT outputs an effective mask file and a file with scaling parameters.



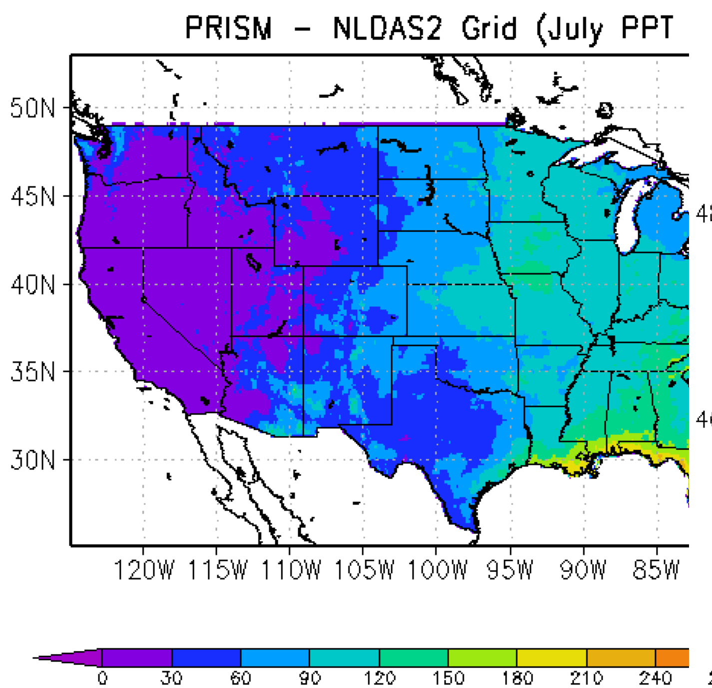


Climate Downscaling Examples

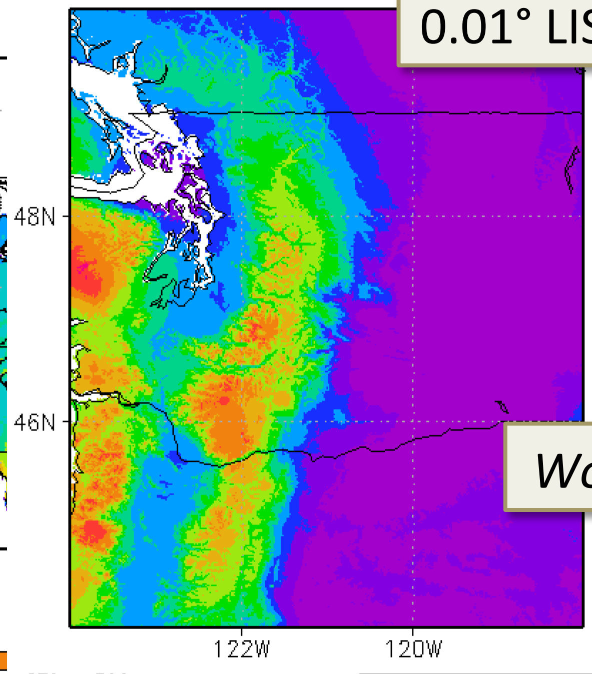
LDT can process and aggregate ~1 KM **PRISM** or **WorldClim** precipitation monthly climatologies to any LSM domain and meteorological forcing source (e.g., AGRMET grid) …

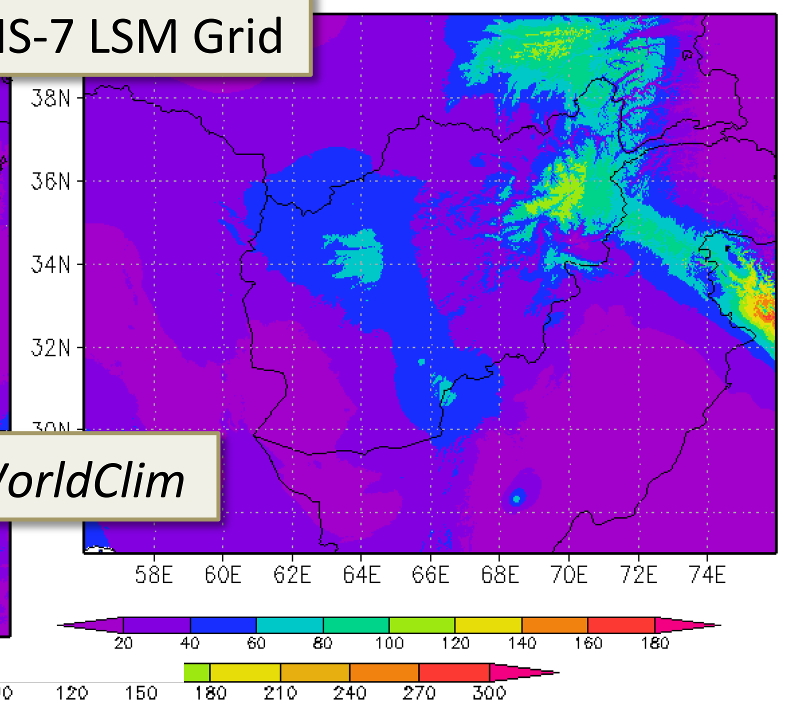


0.25° AGRMET Grid



0.01° LIS-7 LSM Grid



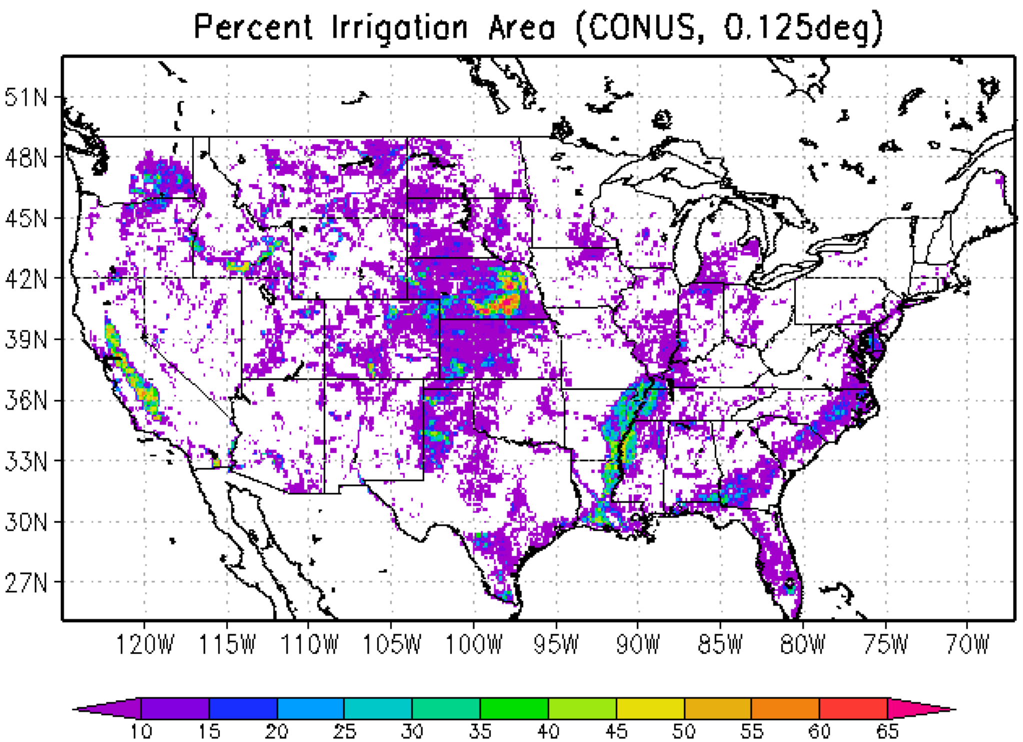


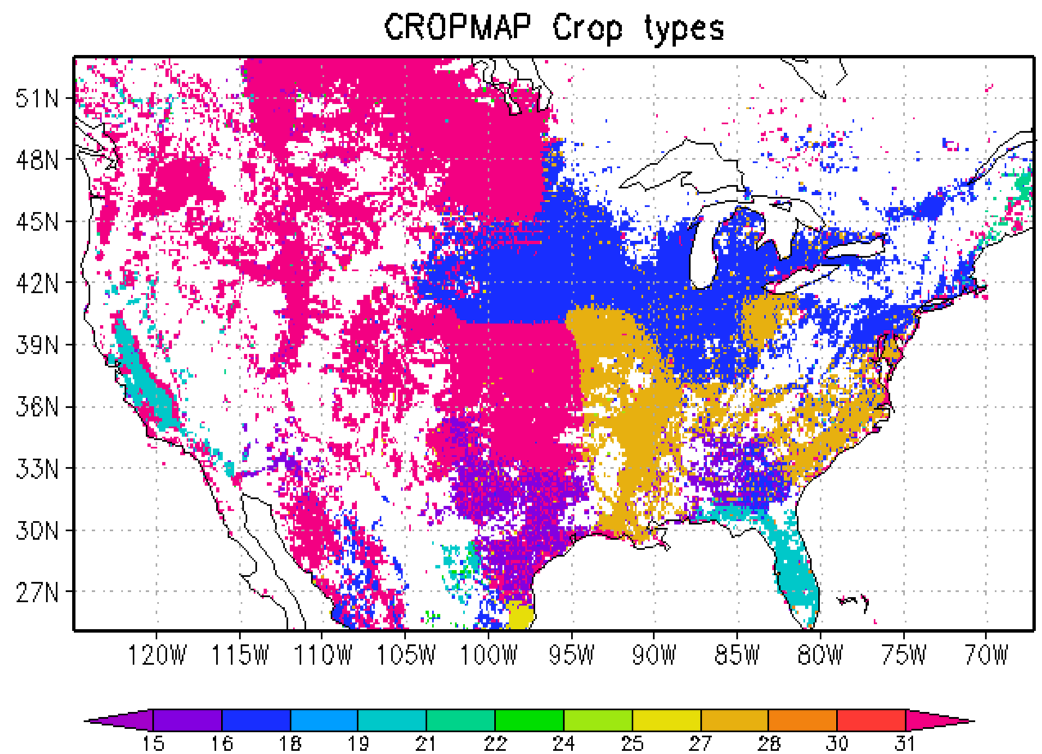
WorldClim

Irrigation and Crop type Examples

MODIS-based irrigation percent for each 0.125 deg gridcell, circa 2001 *(see Ozdogan et al. 2010).*

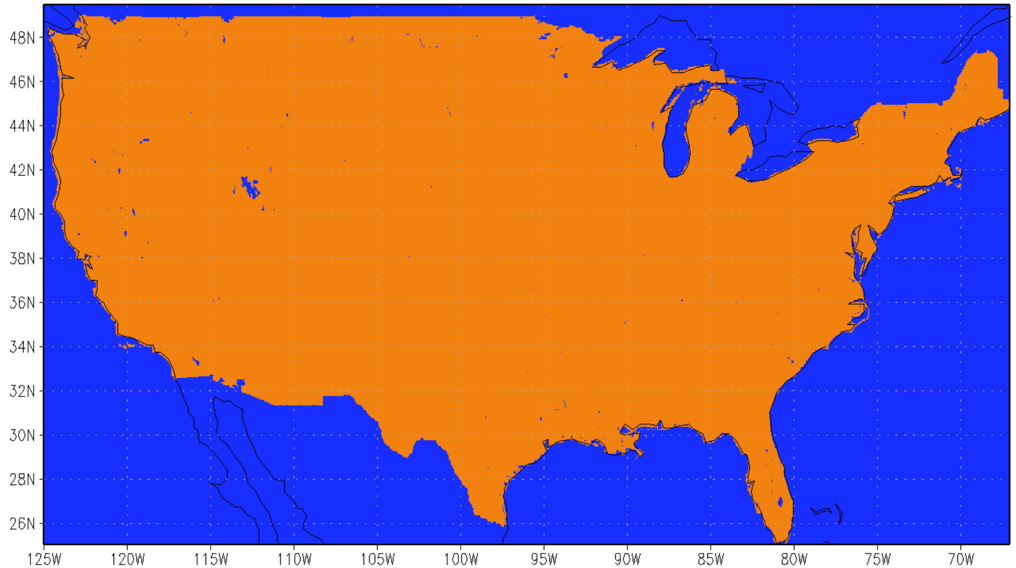
UMD+CROPMAP dominant crop categories derived from crop census databases *(see Ozdogan et al. 2010).*

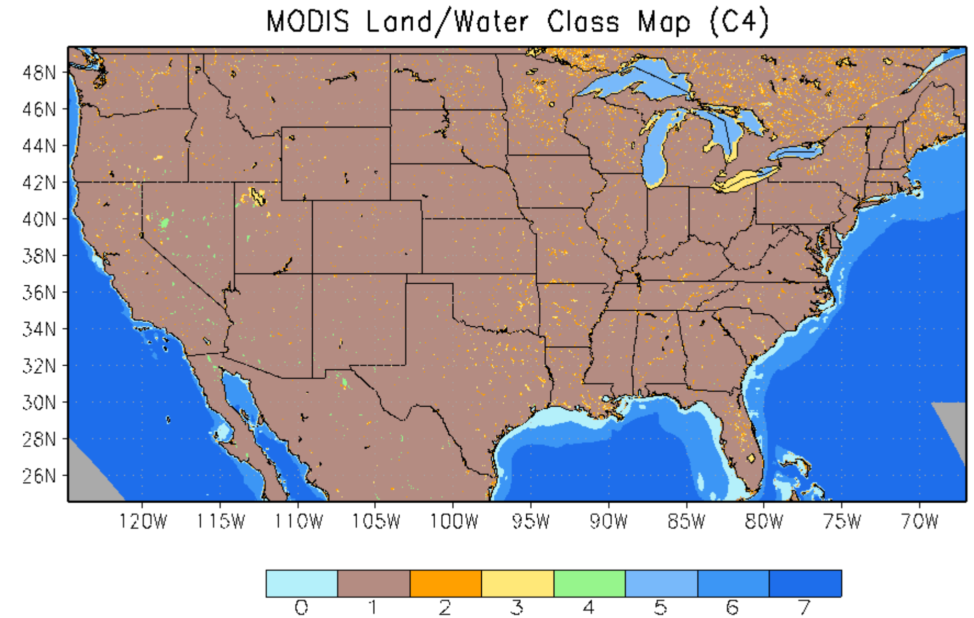


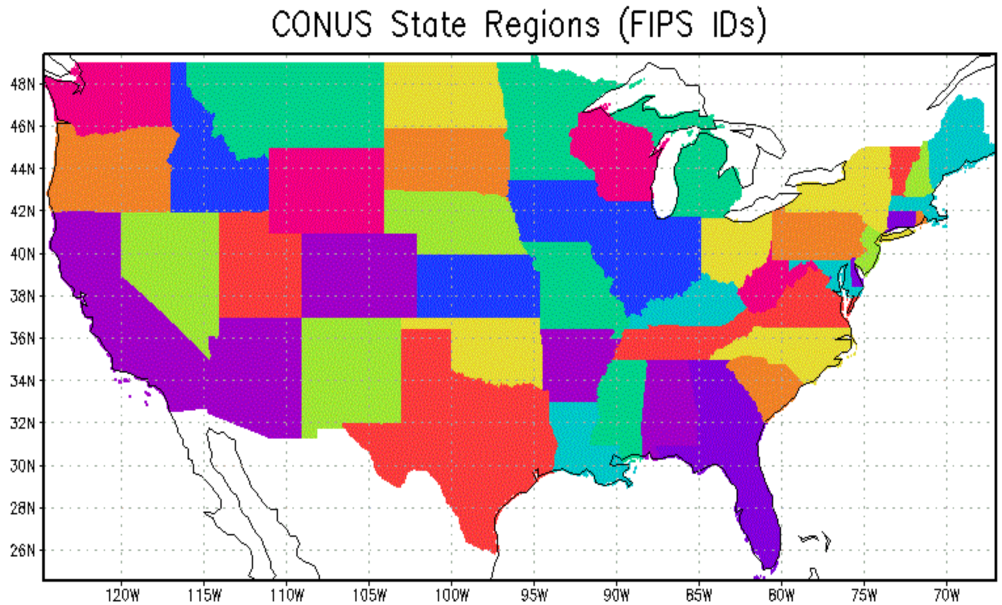


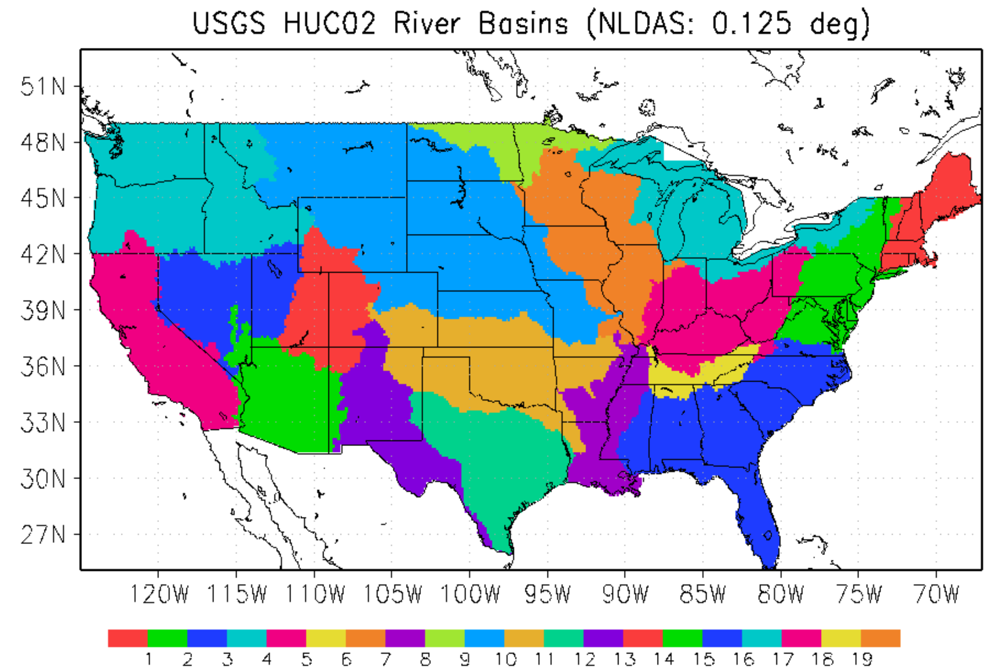
Ozdogan et al., 2010, Simulating the effects of irrigation over the U.S. in a land surface model based on satellite-derived agricultural data, *J. Hydromet.*, 11, 171-184.

Mask (and Regional Mask) Examples



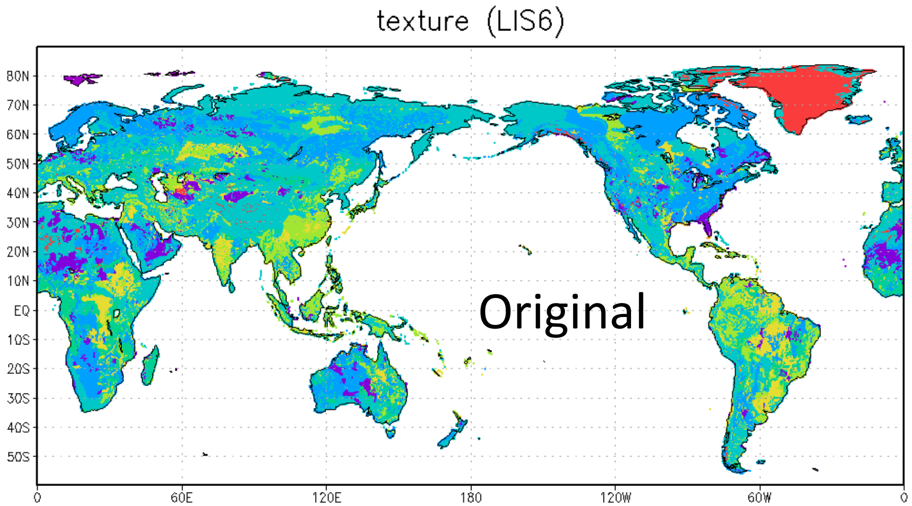


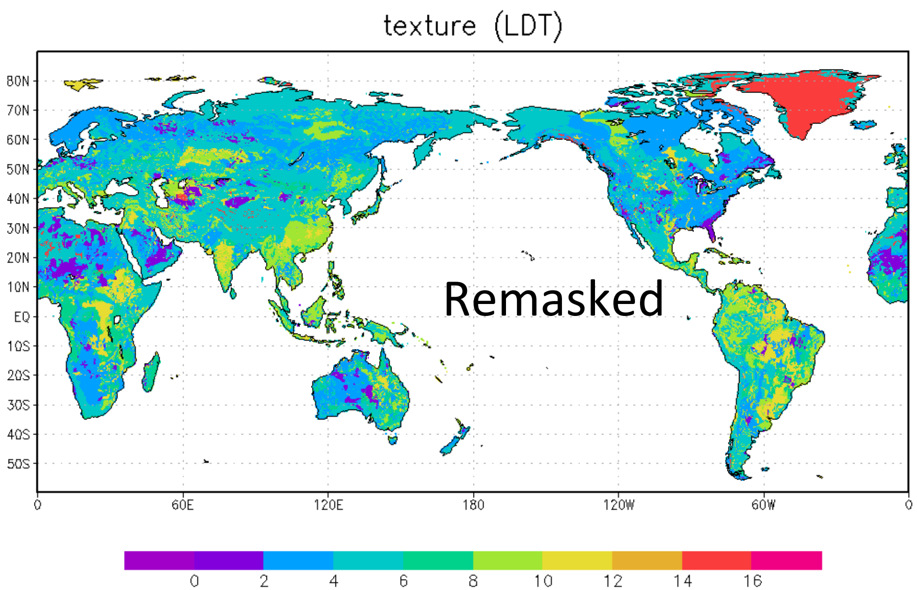




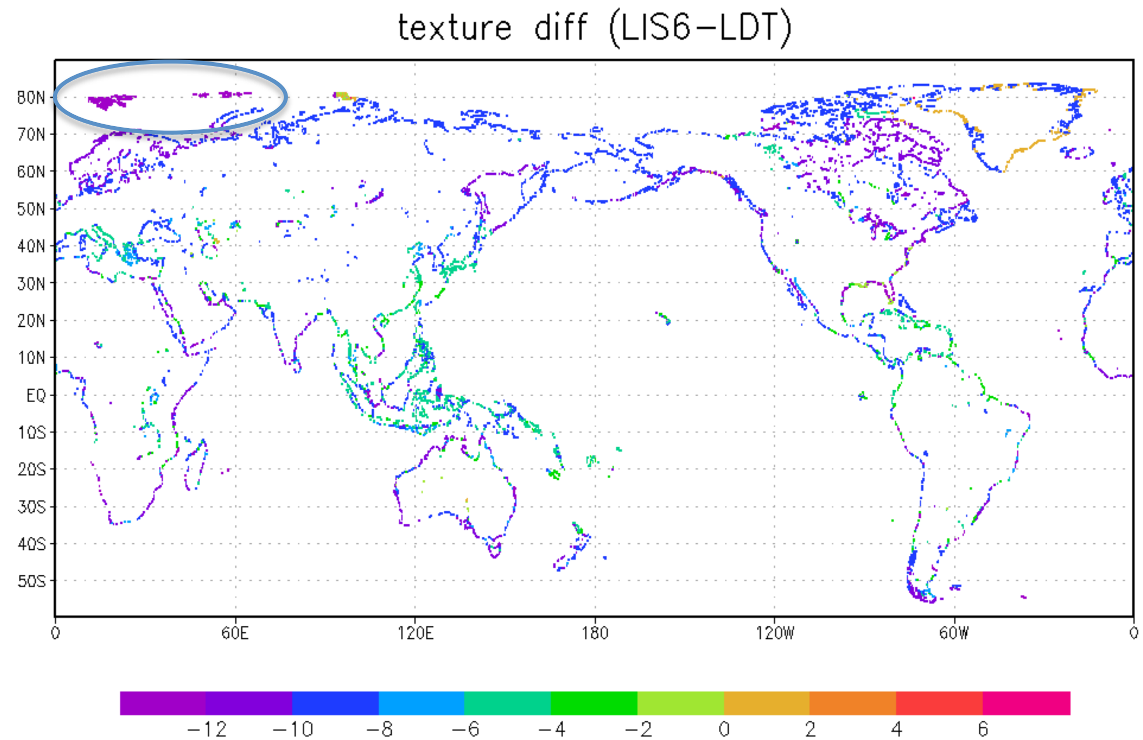
Parameter-Mask Consistency Check Example

LDT ensures consistency between the landmask and parameter files.
For example, LDT modifies the NCEP/NCAR global STATSGO soil texture file's built-in land/water mask to match the UMD land-mask.





Difference between original and re-masked soil texture



Summary

- The Land surface Data Toolkit (LDT) is a new preprocessing toolkit for LIS-7's model parameters and DA inputs.
- LDT offers several features:
  - Multiple parameter processing options;
  - Observation-based DA options (e.g., CDF-matching);
  - Generates ensemble-based restart files
- LDT supports a variety of options, like parameter tiling, and parameter data types, like irrigation maps and lake model data

Future Work

- Add generic capability to bias correct forcing variables (e.g., precipitation).
- The ability to process OPTUE outputs for use in a subsequent LIS run.
- Implement observational correction strategies used (Cressman, OI) into LDT -- for updating snow (and possibly other) data sources.
- Add a layer of machine learning tools (ANN/Bayesian classifier) that will enable the blending of different observational sources (e.g., reprocess LPRM against in-situ data).

Future Work

- Apply HYMAP ("Hydrological Mapping") parameter processing for hydrological modeling applications.
- Implement original LSM parameter preprocessing code (e.g., for CLSM, VIC, etc.).
- Improve computational I/O (e.g., parallel netcdf and other options).
- Replace the spatial interpolation code with ESMF (would be major change also needed by LIS and LVT).

LIS7 capabilities and features

New Land surface models

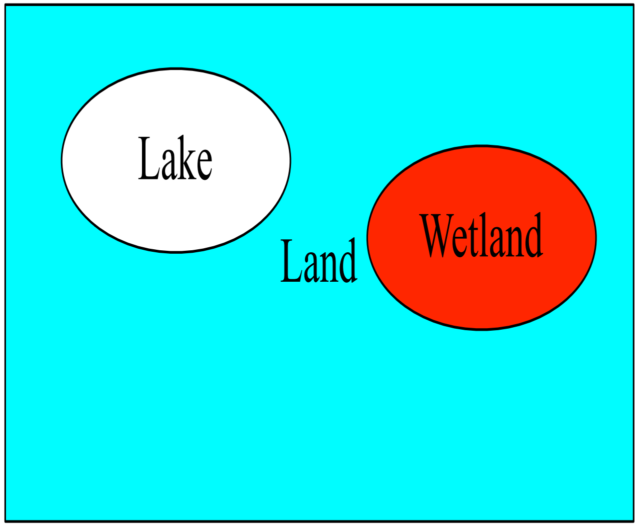
- VIC(4.1.1, 4.1.2)
- CABLE
- Catchment (Fortuna 5.2)
- SAC-HTET
- Noah (2.7.1, 3.3, 3.4, MP)
- GeoWRSI
- FASST
- SiB2, HySSIB
- CLM2
- HTESSEL
- Mosaic
- JULES
- CLM4.5
- SHEELS

- FLake

- Noah-urban

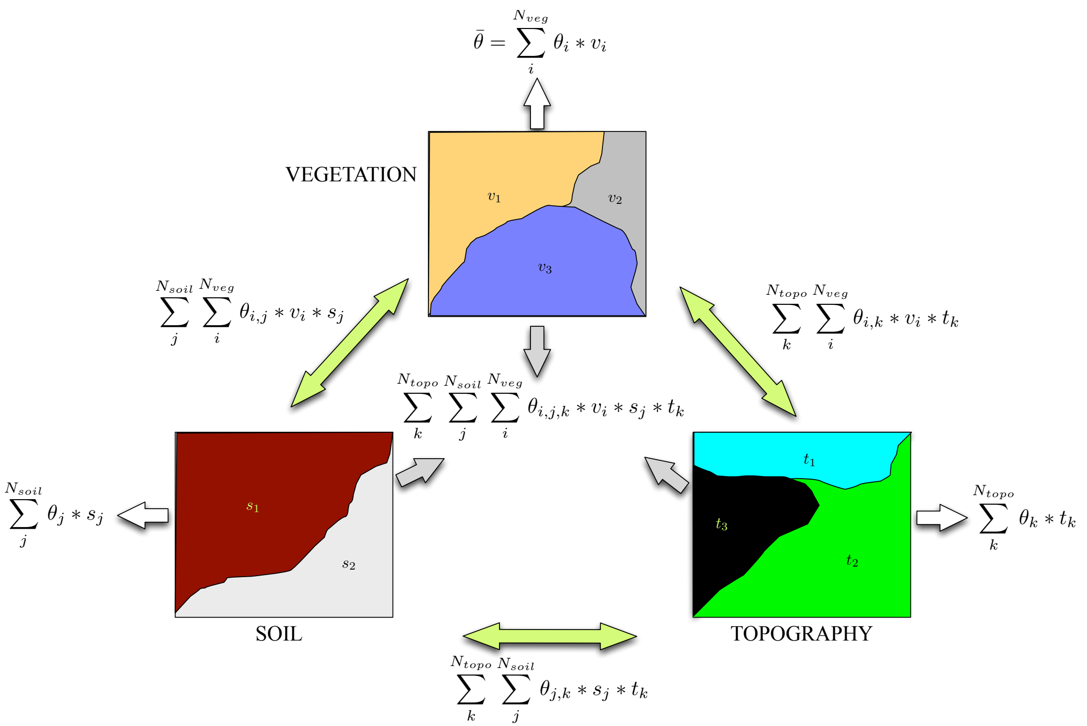
Support for surface models

- LIS7 will allow multiple "surface" models in addition to land surface models.
  - E.g. A domain could consist of land points (running land surface models), lake points (running lake models) and wetland points (running wetland models).
  - LIS would aggregate and "quilt" the outputs from these different model types into a single output structure.
  - LIS7 includes the definition of "patchy domains" that represent the sub-domains that run surface models



Flexible subgrid tiling options

- LIS6 follows the strategy of tiling based on vegetation distribution alone and ignores the sub-grid heterogeneity of other land surface parameters (soils, topography etc.)
- LIS7 allows the tiling space to be determined by other land surface characteristics (e.g. soil texture, elevation, slope, aspect, etc.)
- Depending on the domain, users can select characteristics to define tiling by: (e.g. tile by vegetation and elevation).
- Required data preprocessing done through LDT



Forcings

- Forcing structure has been modified to allow for more flexible data transformations
  - Supports overlays
  - Supports forcing ensembles
  - Allows for online bias-correction
- No more “base” and “supplemental” forcings. Everything is a “met” forcing and they can be overlaid in the order in which the user chooses.
  - LIS6 only allows the overlay of supplemental forcings and not baseforcings.

Data Assimilation

- Includes support for smoothing algorithms (e.g. ensemble kalman smoother)
- Support for multiple data products: AMSR-E, SMMR, SSM/I, ECV, ASCAT, GRACE.
- Options for specifying spatially varying error parameters
- Support for radiance assimilation.
- A new implementation of a fast fourier transform and supports horizontal correlations.

Optimization and Uncertainty Estimation

- The new structure allows for the concurrent parameter estimation across different model classes.
  - E.g. parameter estimation of both LSM and RTM parameters against both soil moisture and Brightness temperature observations.

Routing

- Includes a suite of routing algorithms
  - Source-to-sink methods: NLDAS router, HYMAP
  - Models that includes lateral transport of soil moisture (and feedback to the model states): NDHMS
- Associated topographical processing will be supported through LDT

- A perl-based build system
- Prompts the user for the choice of libraries, compile time options.

```
~/LISv7.0/src.stable % ./configure
-----------------------------------------------------------------------
Setting up configuration for LIS version 7.0...
Parallelism (0-serial, 1-dmpar, default=1):
Use openMP parallelism (1-yes, 0-no, default=0):
Optimization level (-2=strict checks, -1=debug, 0,1,2,3, default=2):
Assume little/big_endian data format (1-little, 2-big, default=2):
Use GRIBAPI? (1-yes, 0-no, default=1):
Grib Table Version (default = 128):
Grib Center Id (default = 57):
Grib Subcenter Id (default = 2):
Grib Grid Id (default = 255):
Grib Process Id (default = 88):
Enable AFWA-specific grib configuration settings? (1-yes, 0-no, default=0):
Use NETCDF? (1-yes, 0-no, default=1):
NETCDF version (3 or 4, default=4):
NETCDF use shuffle filter? (default = 1):
NETCDF use deflate filter? (default = 1):
NETCDF use deflate level? (default = 9):
Use HDF4? (1-yes, 0-no, default=1):
Use HDF5? (1-yes, 0-no, default=1):
Use HDFEOS? (1-yes, 0-no, default=1):
Use MINPACK? (1-yes, 0-no, default=1):
Use CRTM? (1-yes, 0-no, default=1):
Use CMEM? (1-yes, 0-no, default=1):
-----------------------------------------------------
 configure.lis file generated successfully
-----------------------------------------------------
Settings are written to configure.lis in the make directory
If you wish to change settings, please edit that file.
To compile, run the compile script.
-----------------------------------------------------------------------
```

External libraries used in LIS

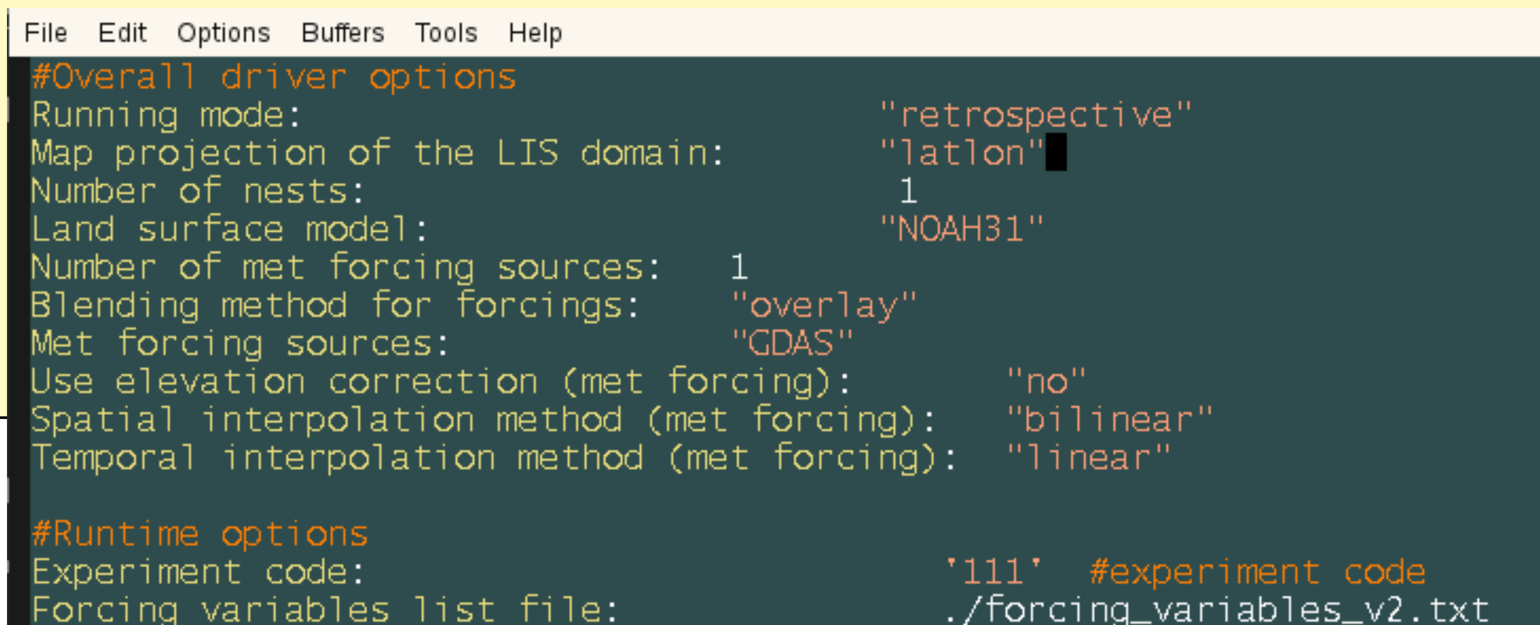
- Uses ESMF5 series – backward compatibility is ensured.
  - Can use newer releases of ESMF without interface/code changes in LIS.
- Grib – consolidated the use of 3 different grib libraries (NCEP, AFWA, NCAR) with the ECMWF developed grib-api library.
  - Includes a documented F90 API – grib-api
  - Supports both grib1 and grib2
  - No need to distribute libraries with LIS
- LIS7 supports NETCDF4 (and NETCDF3) with options for data compression.
  - NETCDF output follows the CF and COARDS conventions.
- HDF5 and HDF4
  - Optional, used only for reading certain remote sensing datasets

Time handling

- LIS7 includes support for 'variable timestepping'
  - Each component (LSM, Forcing, RTM, etc.) sets its own internal timestep.
  - LIS computes the minimum timestep among these components as the timestep for the global clock
  - This enables automatic temporal aggregation of forcings if the LSM is run at a timestep greater than the forcing timestep.
- The use of ESMF-based alarms are eliminated
  - So that synoptic/monthly/weekly alarm intervals can be handled more easily
  - Resetting/Looping of the global clock can be handled more easily.
  - Allows better nesting support while coupling to WRF.

Configuration

- The C-function tables have been changed from an array structure to a linked list structure
  - This eliminates the need for hardcoded array sizes for C-based function tables
  - This also enables the use of strings as keys to store functions in the C-tables. This leads to a more intuitive lis.config interface:
- Eg:



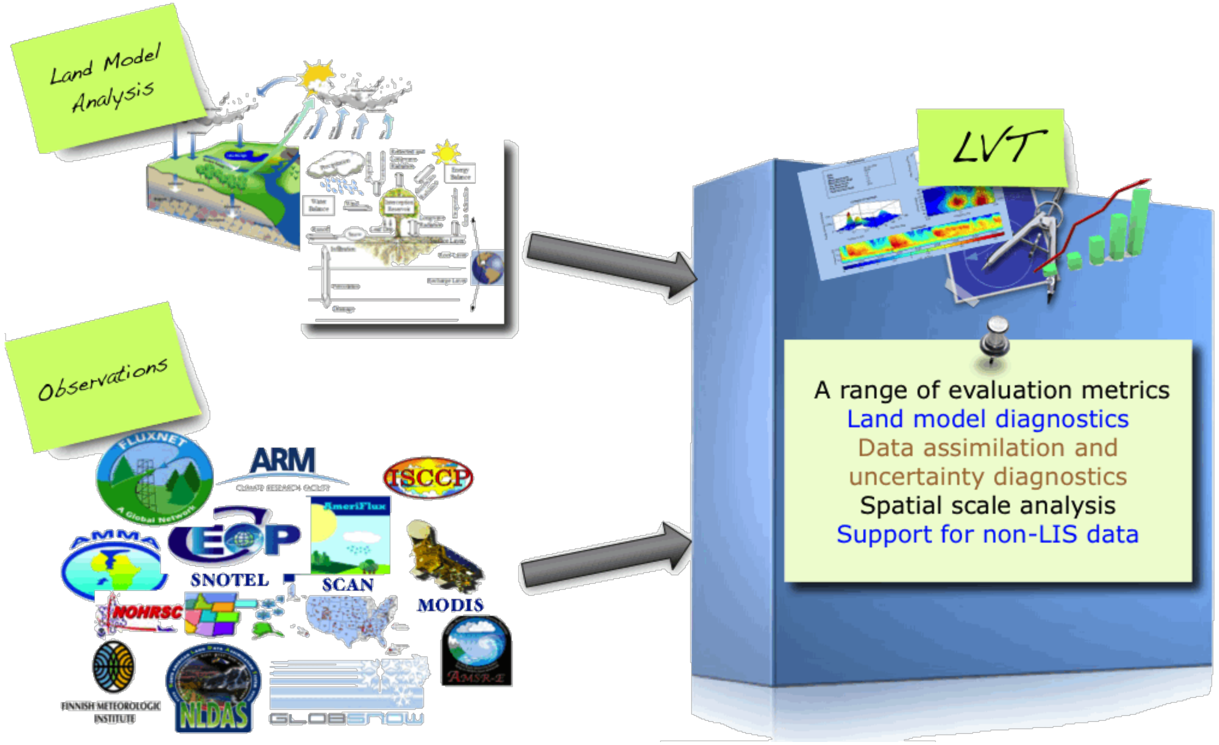
```
File  Edit  Options  Buffers  Tools  Help
#Overall driver options
Running mode:                              "retrospective"
Map projection of the LIS domain:          "latlon"
Number of nests:                            1
Land surface model:                        "NOAH31"
Number of met forcing sources:   1
Blending method for forcings:    "overlay"
Met forcing sources:             "GDAS"
Use elevation correction (met forcing):        "no"
Spatial interpolation method (met forcing):    "bilinear"
Temporal interpolation method (met forcing):   "linear"

#Runtime options
Experiment code:                              '111'   #experiment code
Forcing variables list file:                  ./forcing_variables_v2.txt
```

Misc

- Options for spatial downscaling (slope-aspect correction of radiation, PRISM/WorldClim-based downscaling of precipitation)
- Support for irrigation modeling
- The restart files are written in NETCDF4 formats (as an option)
- Better support for higher compiler optimization levels
- The default 5-level hierarchy of LIS outputs will be changed to a 3-level hierarchy (OUTPUT/MODEL/YEARMONTH)

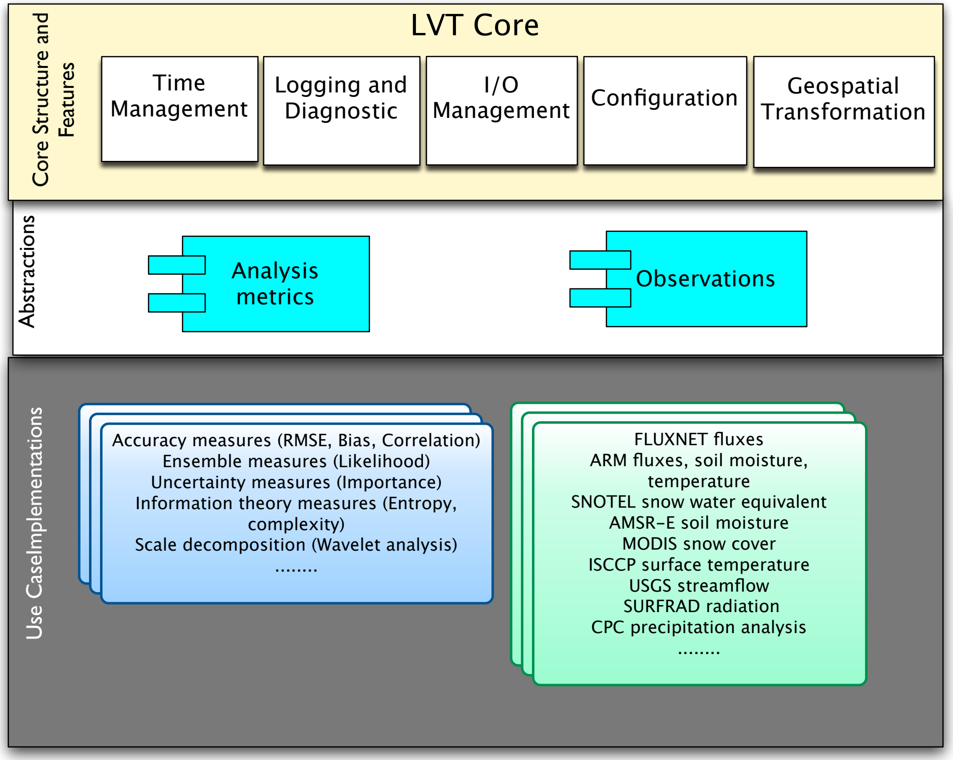
Land surface Verification Toolkit (LVT)



- LVT is a framework developed to provide an automated, consolidated environment for systematic land surface model evaluation
- Includes support for a range of in-situ, remote-sensing and other model and reanalysis products.
- Supports the analysis of outputs from various LIS subsystems, including LIS-DA, LIS-OPT, LIS-UE

Kumar et al. (2012), Land surface Verification Toolkit (LVT) – A generalized framework for land surface model evaluation, Geosci. Model. Dev.

Design of LVT



- Designed as a stand-alone system; Analysis instances are enabled by specifying a configuration file (much like LIS). No external scripting is required.
- Designed as an object-oriented framework with extensible features enabled for
  - Specifying new metrics
  - Specifying new observational datasets.

Observational data support – A growing list

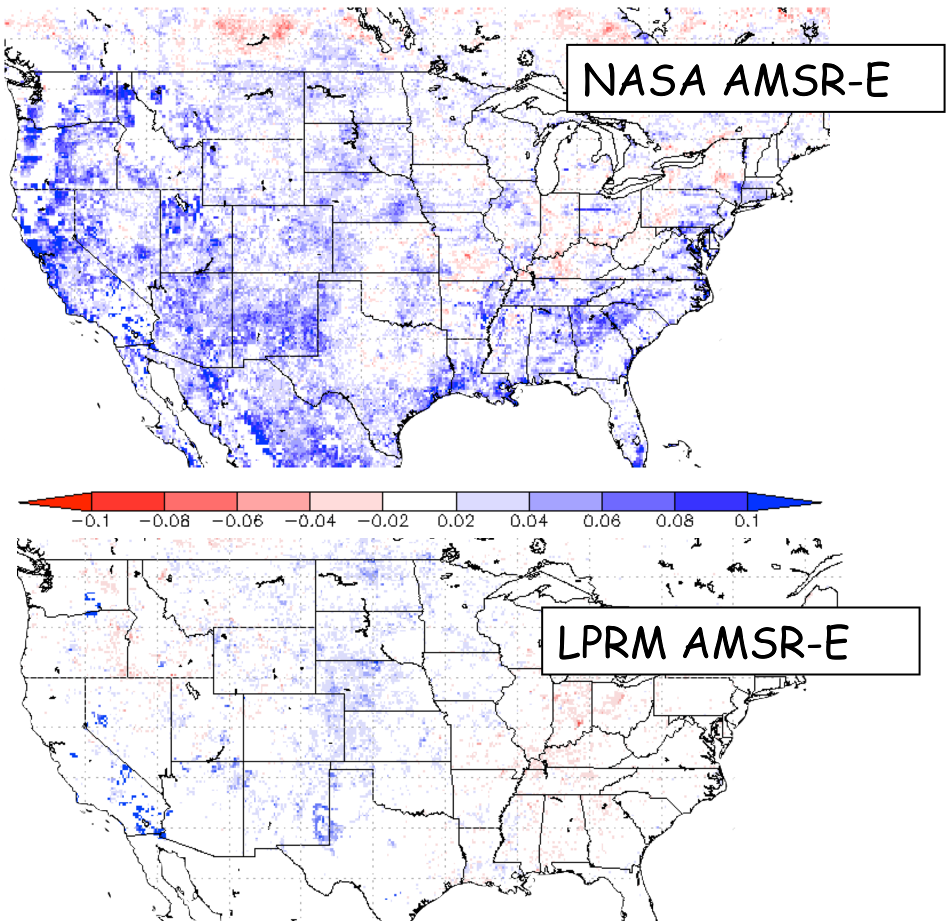
Dataset	Measurement variables
Model/reanalysis outputs	
Agricultural Meteorology Model (AGRMET) from the Air Force Weather Agency (AFWA)	Water and energy fluxes, Soil moisture, soil temperature, Snow conditions, meteorology
NLDAS model outputs Mitchell et al. (2004)	Water and energy fluxes Soil moisture, soil temperature, snow conditions, meteorology
GLDAS model outputs Rodell et al. (2004b)	Water and energy fluxes, Soil moisture, soil temperature, snow conditions, meteorology
Canadian Meteorological Center (CMC) snow depth analysis Brown and Brasnett (2010)	Snow depth
Snow Data Assimilation System SNODAS; Barrett (2003)	Snow depth, snow water equivalent

Satellite and remote sensing data	
AFWA NASA Snow Algorithm ANSA; Foster et al., 2011	Snow cover, snow depth, snow water equivalent
GlobSnow; Pulliainen (2006) (www.globsnow.info/)	Snow cover, snow water equivalent
International Satellite Cloud Climatology Project; ISCCP; Rossow and Schiffer (1991) (isccp.nasa.gov)	Land surface temperature
MODIS/Terra Snow cover 500 m MOD10A1; Hall et al. (2006)	Snow cover
MODIS Evapotranspiration product MOD16; Mu et al. (2007)	Evapotranspiration
NASA Level-3, soil moisture retrieval from AMSR-E (AE_Land3) Njoku et al. (2003)	Soil moisture
Land Parameter Retrieval Model (LPRM) from NASA GSFC and VU Amsterdam Owe et al. (2008)	Soil moisture

In-situ measurements	
AMMA (database.amma-international.org/)	Water and energy fluxes, soil moisture, soil temperature
Atmospheric Radiation Measurement (ARM) (www.arm.gov)	Water and energy fluxes, Soil moisture, soil temperature, meteorology
Ameriflux (public.ornl.gov/ameriflux/)	Water and energy fluxes
Coordinated Energy and water cycle Observations Project (CEOP) (www.ceop.net/)	Water and energy fluxes, soil moisture, soil temperature, meteorology
National Weather Service Cooperative Observer Program (COOP) (www.nws.noaa.gov/om/coop/)	Snow depth, precipitation, land surface temperature
NOAA CPC unified Higgins et al. (1996)	Precipitation
Gridded FLUXNET Jung et al. (2009)	Water and energy fluxes
Finnish Meteorological Institute FMI/SYKE; www.environment.fi/syke	Snow water equivalent
Global Summary of the Day (GSOD)	Snow depth
International Soil Moisture Network (www.ipf.tuwien.ac.at/insitu/)	Soil moisture
Soil Climate Analysis Network (SCAN; www.wcc.nrcs.usda.gov/scan/)	Soil moisture Soil temperature
WMO synoptic observations	Snow depth
NRCS SNOwpack TELemetry network (SNOTEL; www.wcc.nrcs.usda.gov/snow/)	Snow water equivalent
Surface Radiation Network (SURFRAD) (www.srrb.noaa.gov/surfrad/)	Downwelling shortwave, downwelling longwave
Southwest Watershed Research Center (SWRC; www.tucson.ars.ag.gov/dap/)	Soil moisture, soil temperature
USGS water data (waterdata.usgs.gov/nwis)	Streamflow
AMSR-E radiances (mrain.atmos.colostate.edu/LEVEL1C/)	Brightness temperature for different channels

Metrics development in LVT

- A large suite of analysis metrics, including accuracy-based metrics, ensemble and uncertainty measures, information theory metrics and similarity measures has been built into LVT



Change in Metric entropy as a result of the assimilation of soil moisture retrievals of AMSR-E from NASA and LPRM algorithms

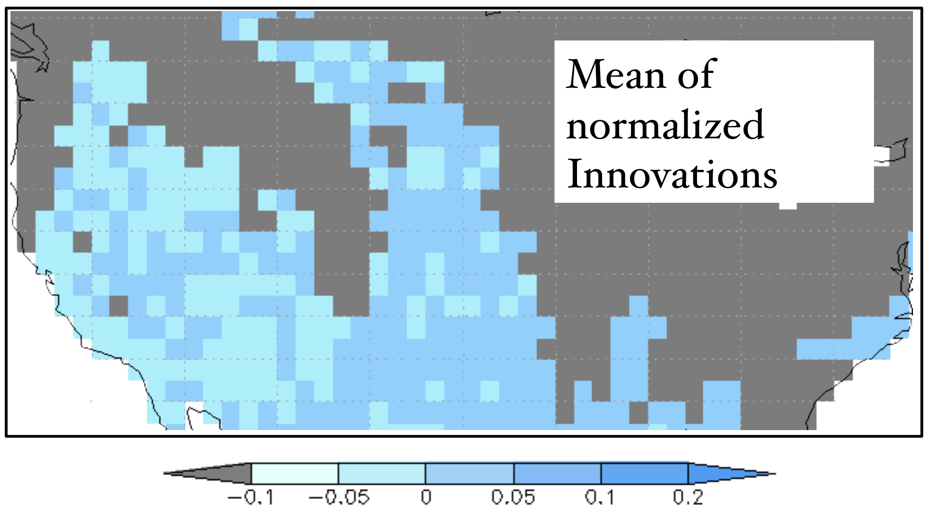
Metric Class	Examples
Accuracy metrics	RMSE, Bias, Correlation
Ensemble metrics	Mean, Standard deviation, Likelihood
Uncertainty metrics	Uncertainty importance
Information theory metrics	Entropy, Complexity
Data assimilation metrics	Mean, variance, lag correlations of innovation distributions
Spatial similarity metrics	Hausdorff distance
Scale decomposition metrics	Discrete wavelet transforms

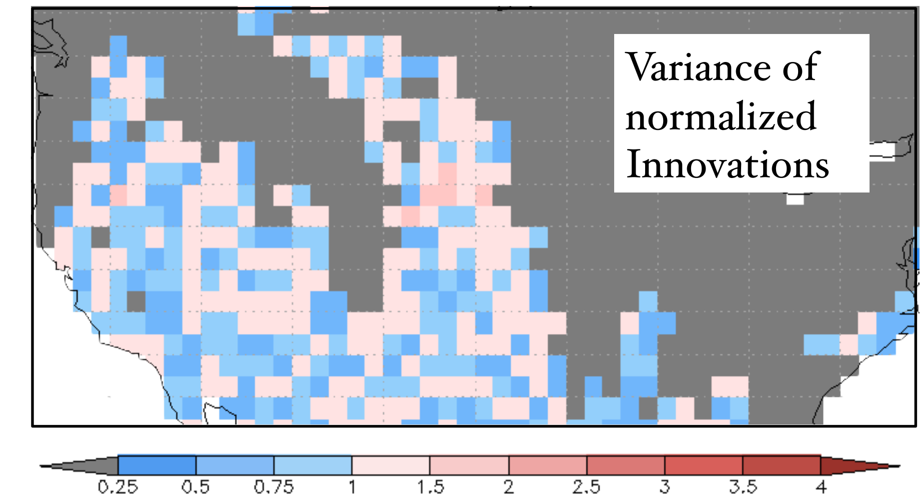
Metric entropy provides a measure of the randomness in the soil moisture time series at each grid point. The availability of information theory metrics in LVT provides a way to discriminate model simulations based on their information content.

Capabilities

- LVT reconciles the differences in spatial and temporal resolutions by bringing the model (LIS) and observational datasets to a common (user-specified) space and time domain.
- Support for datasets in their "native" formats; Once the specific plugin to process a particular dataset is built, datasets can be directly employed within LVT. E.g. ARM-CART measurements.
- Supports non-LIS datasets for intercomparisons – (An observational processing mode in LVT enables the conversion of an external dataset to a "LIS like" form.
- Miscellaneous:
  - Confidence intervals on analysis statistics
  - Analysis outputs in ASCII, binary, GriB, NETCDF formats
  - Probability density functions of computed metrics
  - Stratify analysis by external datasets
  - Stratify analysis based on a model variable (e.g. day-night stratification)
  - Land surface diagnostics

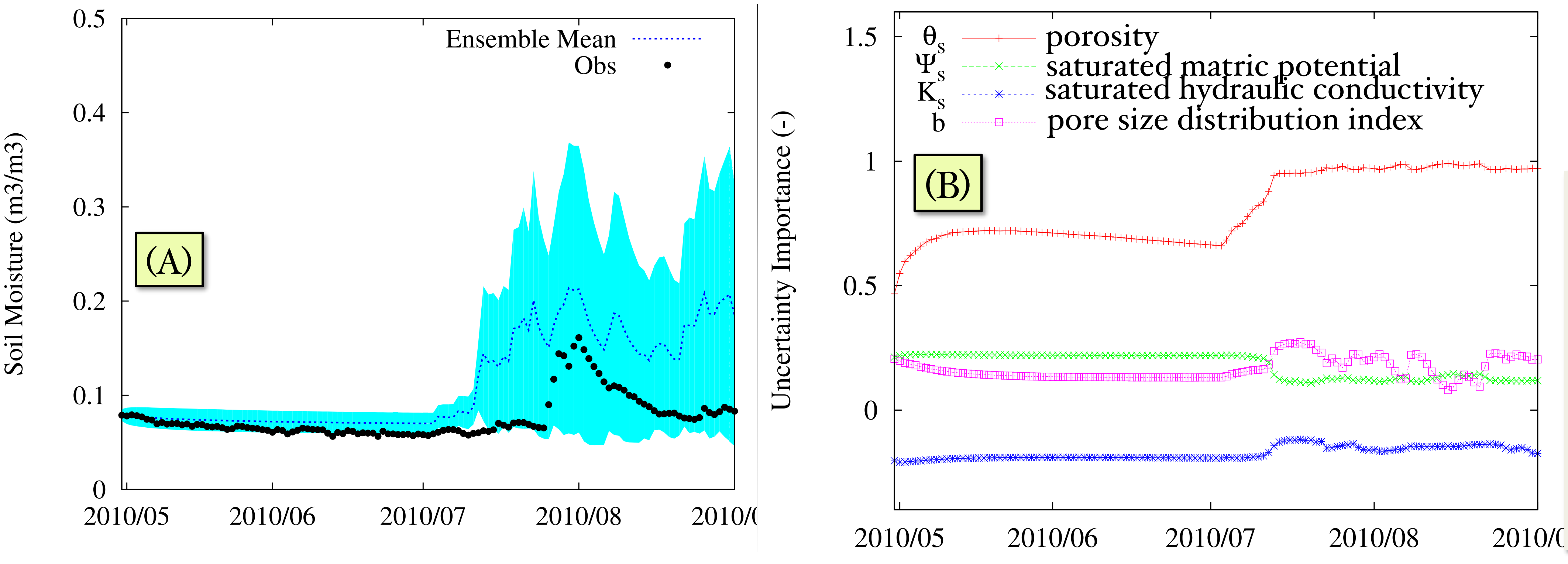
Analysis of LIS-DA outputs





- Deviations from the expected mean and standard deviations of the normalized innovation distribution is used as a measure of the optimality of the data assimilation configuration.

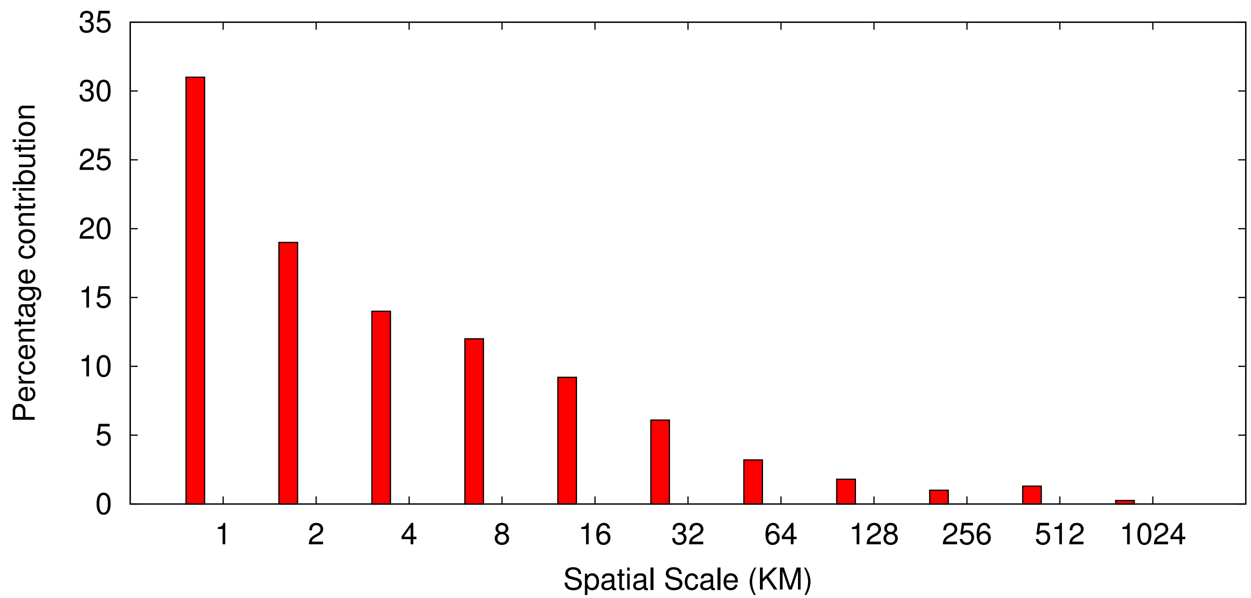
Analysis of LIS-UE outputs



- Uncertainty importance measure: An assessment of the relative contribution of each parameter to the ensemble spread, computed as the correlation between the simulated variable and the the parameter, across the ensemble.

Scale decomposition features

- Tools to characterize the impact of spatial scale on different process variables
- E.g. Discrete Wavelet transforms, spatial similarity measures



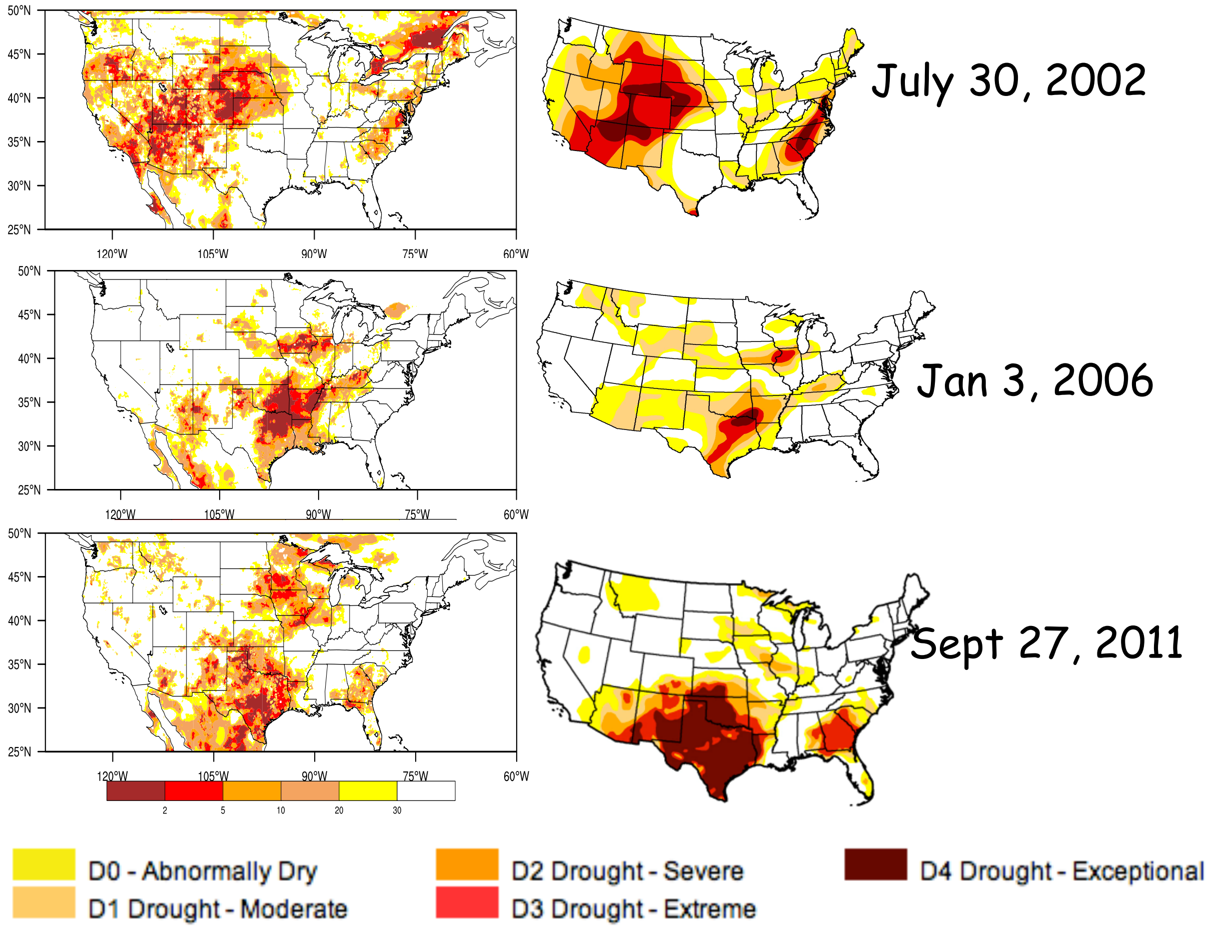
- Percentage contribution to the total improvement in snow covered area POD at different spatial scales, generated by a two-dimensional discrete Haar wavelet analysis.

Hydrological Products development

- A suite of common, normalized indicators used for drought monitoring has been developed in LVT (e.g. Standardized precipitation index (SPI), Standardized Runoff Index (SRI), Standardized Soil Water Index (SSWI), Percentiles

Root zone soil moisture based drought percentiles generated by LVT from a LIS simulation

U.S. Drought monitor estimate



The capabilities of LVT enable an environment for performing systematic evaluation of the OSSEs using various metrics including end-use oriented measures.

Summary

- An environment for the systematic, comprehensive and integrated verification of land surface models with a large suite of metrics.
- LVT supports the outputs from various LIS subsystems including DA, OPT, UE, RTM etc.
- Extensible features for incorporating new metrics and observation sources.
- A conduit for developing hydrological products (e.g. drought/flood indicators).

Questions?